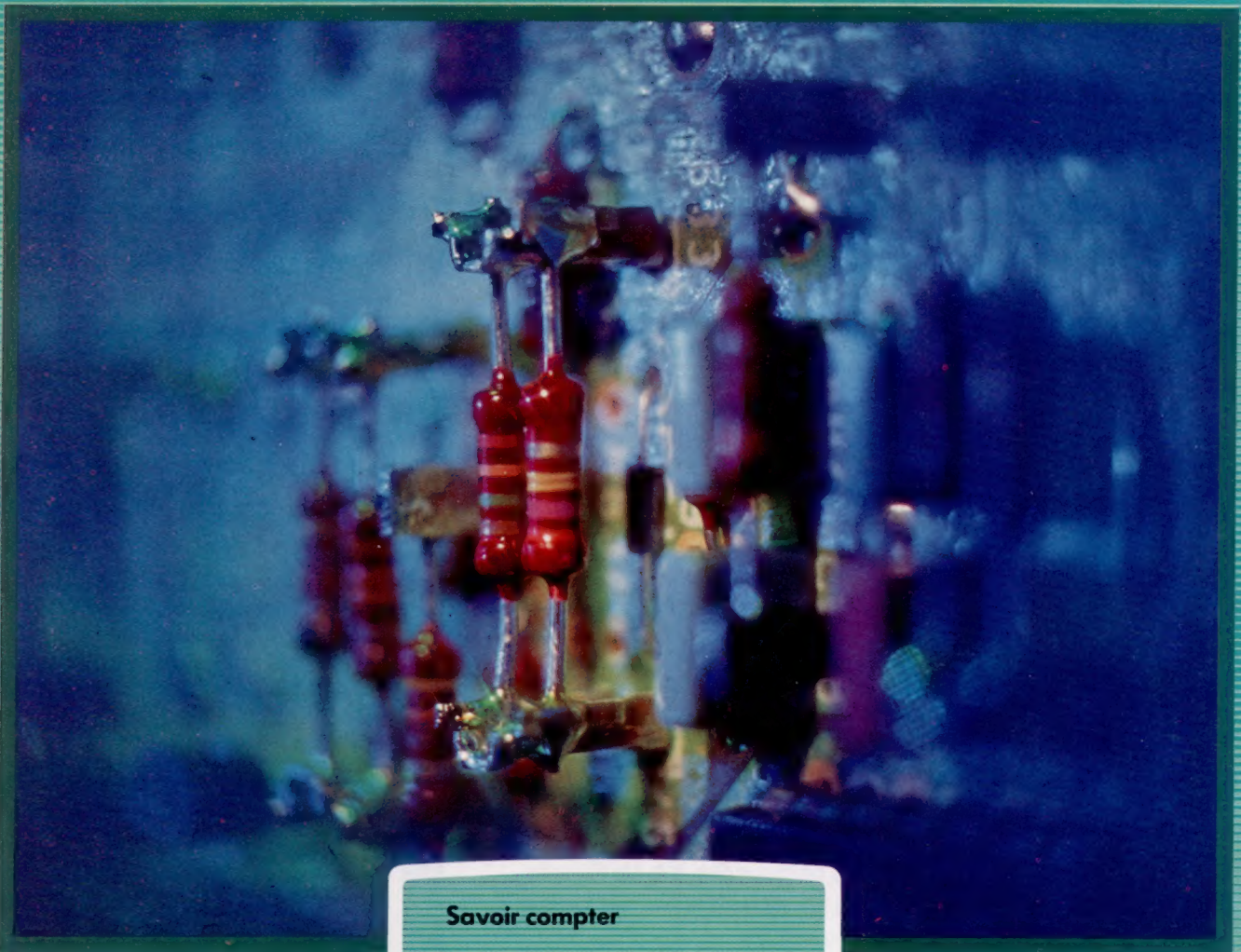


abc

N° 59

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Savoir compter

Massacre au Dragon

Opérations de suspensions

Commodore et variable

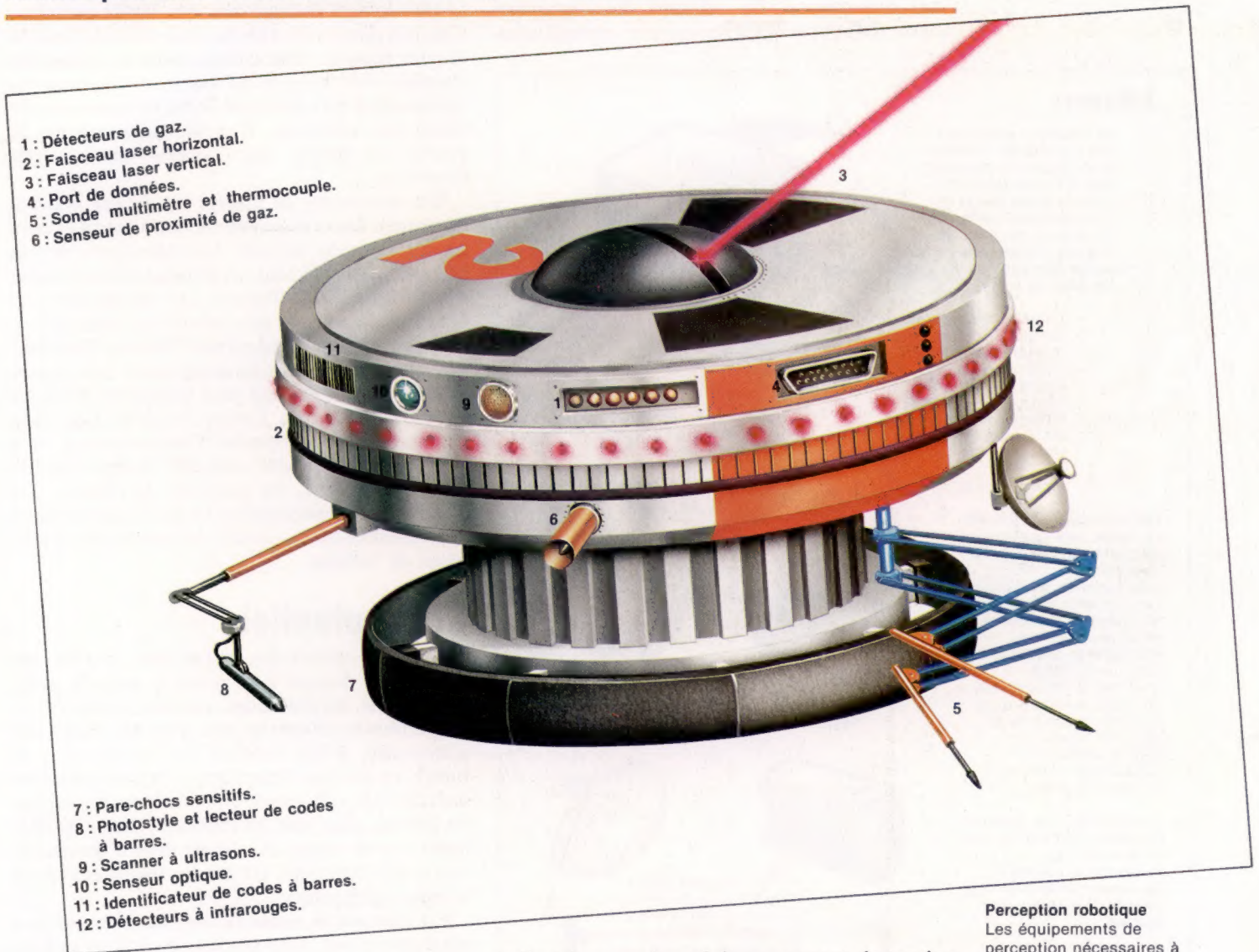
EDITIONS
ATLAS

**Page manquante
(publicité et colophon)**



Sens et sensibilité

Nous avons vu jusque-là les méthodes de contrôle des mouvements des robots. Nous nous intéressons maintenant aux manières dont les robots peuvent « sentir » le monde qui les environne.



Nous avons vu comment un robot se déplace, mais il nous faut également le munir d'un système sensoriel pour lui permettre d'agir de manière indépendante. Il serait tentant d'essayer de concevoir un robot doté de tous les sens humains, de le créer à notre image : il pourrait alors percevoir le monde de façon semblable à la nôtre. Pour l'instant, c'est irréalisable. Les problèmes d'élocution et de vision pour les robots sont tellement complexes qu'ils seront traités spécialement. Nous nous bornerons ici à de simples approximations de la vue et de l'ouïe, qui n'ont rien à voir avec le niveau de complexité de nos propres sens.

Un robot peut « voir » par l'intermédiaire d'un senseur lumineux, généralement une cellule pho-

toélectrique, qui produit un courant de tension variable, selon l'intensité lumineuse. Il s'agit d'un senseur visuel très sommaire, mais susceptible d'être utilisé avec profit. Par exemple, un robot peut ainsi revenir à son point de départ en se dirigeant sur une lumière vive, tout comme il peut suivre une ligne lumineuse. Cette simple cellule photoélectrique permet au robot d'effectuer plusieurs tâches. Un robot travaillant sur une chaîne de montage peut détecter l'absence d'un objet qui lui est nécessaire, par la modification de l'intensité lumineuse. La détection peut être facilitée en accentuant le contraste lumineux déterminant la présence ou l'absence de l'objet par un éclairage spécifique. Le robot peut détecter des modifications de couleur lorsqu'il comporte trois cellules sensibles à des lumières de couleurs diffé-

Perception robotique

Les équipements de perception nécessaires à un robot dépendent entièrement de ses fonctions : plus il aura de senseurs, plus il sera performant. Le robot montré ici présente pratiquement tous les senseurs existants, mais il est peu probable qu'un seul robot mette en œuvre une gamme aussi complète.

(Cl. Steve Cross.)



rentes : rouge, vert et bleu suffiraient à couvrir le spectre visible. Un tel robot pourrait être alors programmé pour retirer des briques, rouges par exemple, d'un tas de briques de diverses couleurs. Ce simple senseur donne ainsi l'impression que le robot a un comportement intelligent.

Un robot équipé d'un micro peut capter des signaux acoustiques. Il ne sera pas capable pour autant d'interpréter ce qu'il entend, mais cela est sans importance. En utilisant des répétitions sonores d'ordres spécifiques pour chaque com-

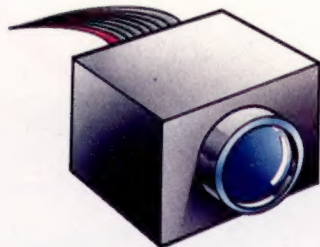
mande, le robot est capable d'établir des identifications sonores opérationnelles. Le nombre de commandes ainsi reconnues est limité, mais nous pourrions de la sorte dire au robot « avance », « arrête », « va à gauche », et ainsi de suite.

On peut également donner au robot, jusqu'à un certain degré, le sens du toucher. Des micro-interrupteurs peuvent être incorporés au robot, de telle sorte qu'ils établissent des connexions électriques lorsqu'ils recevront une légère pression. Cela n'a rien à voir avec la perfection du sens du toucher humain, mais cela peut être très utile. Par exemple, des senseurs de toucher montés sur les extrémités d'un robot mobile lui permettraient de réagir aux obstacles. Il serait alors capable de revenir en arrière pour essayer une nouvelle direction.

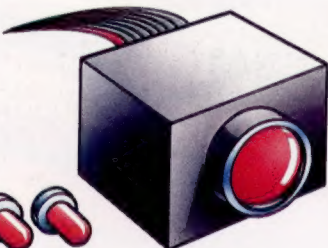
Des détecteurs de fumées ou de gaz peuvent également être mis en œuvre et donner au robot un certain sens olfactif. Les détecteurs de gaz utilisent généralement un élément sensoriel (par exemple, un fil de platine), qui est sensible à la présence de certains gaz, modifiant ainsi le courant électrique qui passe dans l'élément détecteur. Les détecteurs de fumées comportent deux chambres, l'une fermée, qui joue le rôle de référence ou de contrôle, et l'autre ouverte. Les deux contiennent des particules d'hélium ionisé, et le nombre de particules chargées change dans la chambre ouverte, en présence de fumées. Un détecteur qui comptabilise les particules dans les deux chambres est capable de déterminer la présence de fumées.

Senseurs

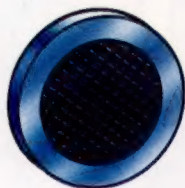
Le senseur optique est une caméra de télévision monochrome à balayage lent, à basse résolution. Il produit une image qui comporte suffisamment d'informations pour de simples tâches telles que suivre une ligne ou détecter un angle.



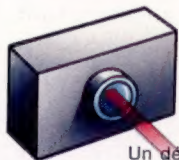
La caméra à infrarouges compose son image de manière très similaire à celle d'une caméra de télévision, mais elle détecte les rayons à infrarouges à la place du spectre lumineux visible.



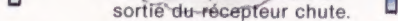
Les ultrasons constituent un rayon sonore de haute fréquence utilisé ici pour la détection directionnelle d'un objet. Le scanner consiste en un émetteur d'ultrasons et en un récepteur-microphone directionnel. Lorsque le rayon d'ultrasons rebondit sur un objet, la nature de la surface réfléchissante déforme l'onde d'écho selon une « signature » unique et reconnaissable.



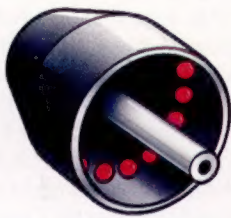
L'analyseur-laser à basse puissance est utilisé pour la détection et l'orientation précises. La lumière laser peut être dirigée de manière très précise.



Un détecteur de fumée consiste en une source lumineuse pointant sur un récepteur photovoltaïque. Lorsque de la fumée et le rayon interfèrent, le résultat en sortie du récepteur chute.



Le détecteur de proximité de gaz consiste en un émetteur de gaz et en un senseur de pression. L'émetteur injecte du gaz dans une chambre, provoquant une élévation de la pression. Lorsqu'un objet est près de l'orifice de la chambre, la pression tombe de manière mesurable.



Le photostyle peut être utilisé pour diverses saisies numériques, dont les codes à barres sont un exemple manifeste.



La sonde multimètre permet de mesurer la distance, la capacité, la tension et le courant, et peut être utilisée en tant que thermocouple pour mesurer des températures.



Gain potentiel

Dans l'état actuel des recherches, il n'est pas possible de donner à un robot le sens du goût. Néanmoins, les méthodes que nous avons décrites permettent d'ores et déjà à un robot de voir, d'entendre, d'être sensible au toucher et à la fumée, ce qui peut s'appliquer à la détection d'un incendie. Un robot peut donc détecter le feu dans un édifice, aller vers les flammes, éviter les obstacles sur sa route, et, s'il est doté d'un extincteur à son extrémité, arroser les flammes avec de la neige carbonique.

En limitant les robots à certains des sens que nous possédons, nous perdons beaucoup de leurs potentialités. Il n'y a aucune raison de limiter le robot à notre propre perception. Une meilleure approche serait de se demander de quels sens peut être doté un robot, et de voir ensuite si ces moyens de perception ont une raison d'être. Un bon exemple concerne les bras de robots.

Supposons que nous voulions qu'un robot déplace un objet. Une manière d'y arriver est de mettre en place des butées autour du bras, de sorte qu'il ne puisse se déplacer que sur une certaine distance dans toutes les directions. Le bras tournerait alors jusqu'à heurter une butée, et, si tout a été positionné de manière appropriée, la main se trouverait alors exactement au-dessus de l'objet à saisir. Une fois celui-là saisi, la main pourrait tourner dans l'autre sens et se déplacer jusqu'à ce que d'autres butées lui fassent savoir



qu'elle doit déposer l'objet. C'est un exemple simple qui montre comment les robots peuvent utiliser des moyens de perception ne relevant pas des sens humains.

Un meilleur exemple est peut-être l'utilisation par le robot de la « vision ». L'homme ne peut percevoir que la lumière visible, une grande partie du champ électromagnétique reste invisible à l'œil humain. Mais il n'y a aucune raison de confiner les robots à nos limites. Des détecteurs à infrarouges peuvent être mis en place dans les cellules photoélectriques; cela permet de mesurer la quantité de chaleur émise par un objet. Les robots industriels peuvent mettre à profit ces détecteurs pour s'éloigner des objets dangereusement brûlants. Mais un robot peut également percevoir la chaleur dégagée par un être humain : votre robot personnel pourrait ainsi être programmé pour venir à votre rencontre et vous accueillir lorsque vous rentrez chez vous le soir ! Les robots sont également capables de détecter les champs magnétiques. Nous l'avons déjà vu à propos des robots qui suivent une trace sur le sol, mais cela peut également servir pour des applications où les robots doivent faire la distinction entre des matières magnétiques et des matières non magnétiques.

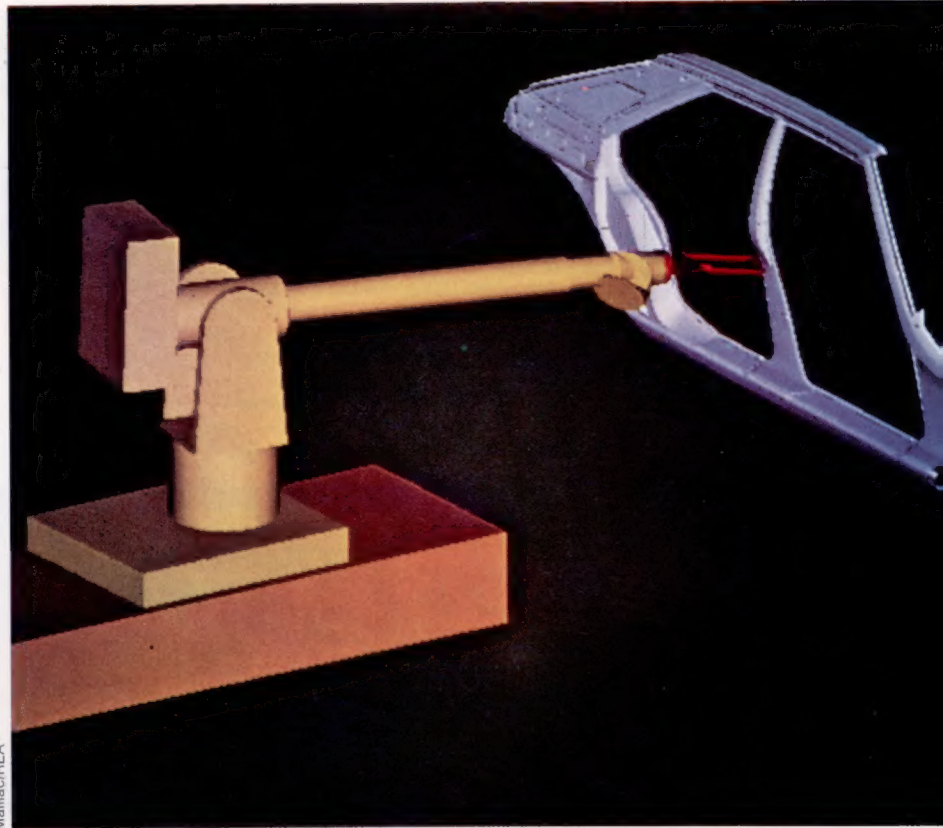
Les senseurs de proximité n'ont pas d'équivalents directs pour nos sens : il s'agit de dispositifs capables de détecter la proximité d'un objet. L'homme utilise à cette fin la vue et le toucher, mais les senseurs de proximité conviennent parfaitement aux applications robotiques. Ils fonctionnent selon divers principes. L'un d'eux utilise un jet d'air propulsé dans une canule : tout objet rencontré renvoie l'air en arrière. Cela a pour effet de créer une rétropression détectable à la variation de pression, avertissant le robot de la proximité d'un objet.

Un autre type de dispositif se fonde sur le fait qu'un circuit électrique avec une capacitance électrique se comporte de manière différente à l'approche d'un autre objet. Une « fuite » électrique entre la capacitance et l'objet (qui possède sa propre capacité) informe le robot de la proximité de l'objet.

Transmetteurs

Il existe en outre des senseurs qui émettent un signal à ultrasons et qui reçoivent l'écho retransmis par l'objet situé à proximité. Le temps de latence entre le signal et l'écho donne la mesure exacte de la distance de l'objet. C'est une méthode qui rappelle celle utilisée pour les allures en navigation ou les systèmes de mise au point automatique « auto-focus » d'appareils photographiques.

Les senseurs à laser dirigent un rayon sur un objet qui le renvoie. La comparaison des deux rayons permet de déterminer la distance de l'objet avec une précision étonnante. Cette technique est utilisable sur de grandes distances. Lors du premier alunissage, un réflecteur fut placé sur le sol lunaire, de sorte qu'un senseur à laser put mesurer avec une extrême précision la distance de la



Terre à la Lune. La précision est, paraît-il, de l'ordre de 15 cm sur une distance de 384 400 km !

Les senseurs de forces mécaniques permettent d'obtenir une information tactile par des moyens beaucoup plus sophistiqués que ceux des micro-interrupteurs mécaniques. Leur principe est de mesurer la modification des propriétés électriques d'un cristal piézo-électrique, lorsqu'il est soumis à une pression, ou en calculant la modification de conductibilité de granules de carbone graphite sous pression. Une autre solution est de mettre en œuvre un indicateur de torsion pour mesurer les forces mécaniques importantes en détectant des modifications dans la résistance électrique d'un fil lorsqu'il est déformé.

Ces senseurs propres aux robots sont connus sous le nom de « transmetteurs ». Ils relèvent une mesure sous une certaine forme (lumière, son, pression), et la convertissent en une autre forme représentative de la première. Sur un robot piloté par ordinateur, les transmetteurs convertissent invariablement les relevés en un signal électrique qui peut être binaire (c'est-à-dire qui est susceptible d'être dans deux états, présent ou absent), ou analogique (le signal varie avec la mesure). Dans ce dernier cas, le signal électrique doit être converti dans une forme que l'ordinateur peut comprendre, en utilisant un convertisseur analogique/numérique.

Il faut bien convenir que les moyens de perception de l'ordinateur, ses sens en quelque sorte, ne sont pas, dans l'état actuel des choses, aussi complets et efficaces que ceux de l'homme. Mais l'ordinateur dispose de davantage de moyens de perception et d'analyse, qui ne cessent de s'affiner.

Dépourvu de sensibilité et de sensations

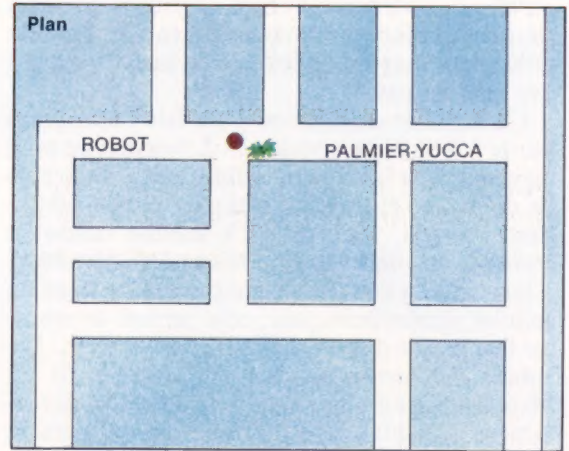
Ce bras de robot industriel nettoie des moulages au sortir du moule, alors qu'ils sont encore trop brûlants pour être manipulés par l'homme. Le robot est insensible à la chaleur, bien sûr, et effectue donc le travail beaucoup plus rapidement.



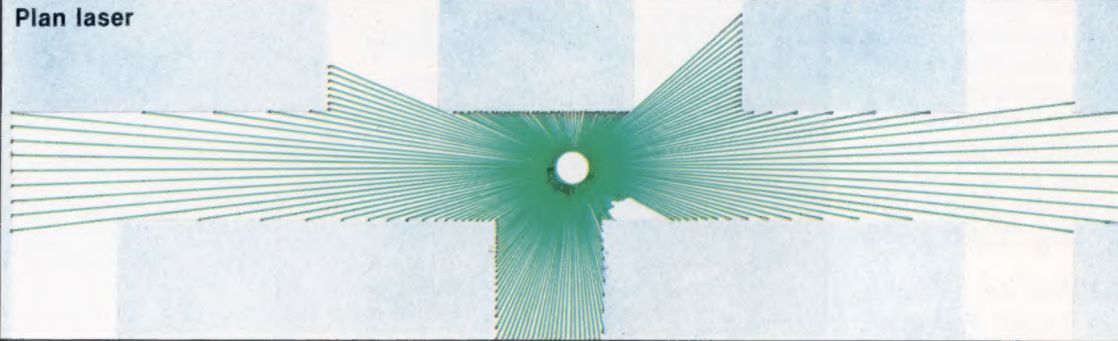
Le sens des capteurs

Le plus grand problème des robots est de prendre la mesure du monde qui les entoure. Pour cela, l'augmentation du nombre des capteurs et l'amélioration de leurs performances peuvent répondre à la question. Car un seul capteur ne donnera jamais une information complète de l'image de son environnement. Plusieurs capteurs donnant des informations contradictoires seront même nécessaires.

Le plan ci-contre montre l'exemple d'un robot se déplaçant dans des couloirs peints en blanc. Une seule source de lumière existe. La lumière réfléchiée par un mur dépend donc de l'orientation de celui-ci. Près du robot, se trouve un palmier-yucca.

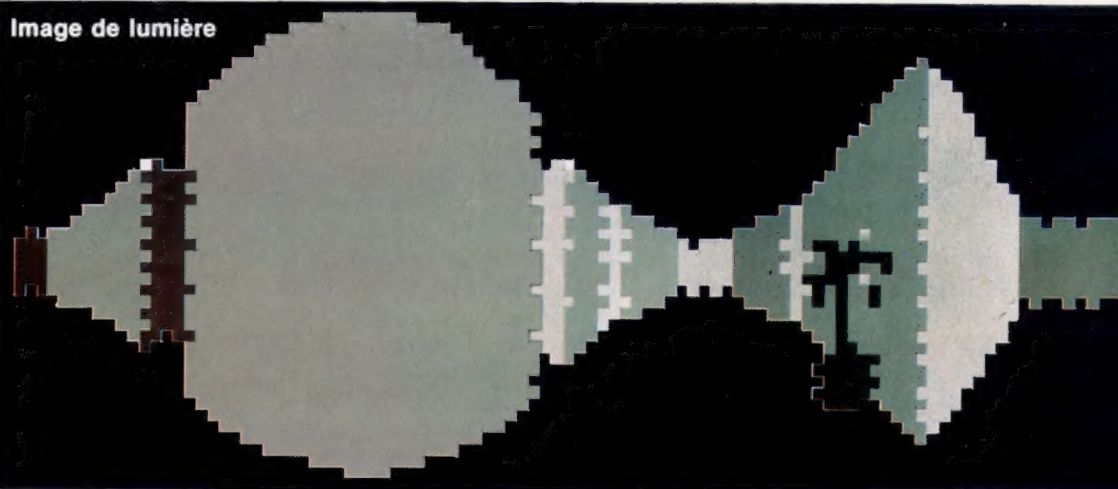


Plan laser



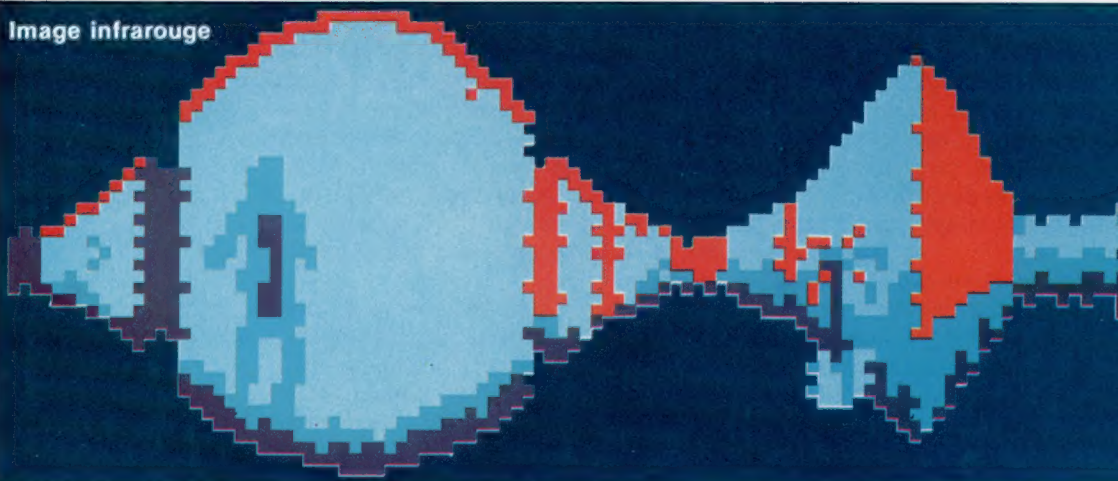
Les rayons d'un laser permettent au robot de se représenter un plan assez précis de l'environnement et révèlent les contours du palmier-yucca. Un léger mouvement du robot produira un effet de parallaxe par rapport au palmier-yucca. De là, le robot distinguera le palmier-yucca comme un objet isolé entre les murs.

Image de lumière



Une caméra de télévision donne une image monochrome basse-résolution de l'espace. Cela permet à un analyseur d'images de détecter des reliefs et d'entraîner des mouvements. Mais c'est insuffisant pour se rapprocher correctement du palmier-yucca.

Image infrarouge



L'image infrarouge est aussi peu définie que celle d'une image cathodique, mais elle révèle que la température du yucca est différente de celle de son environnement immédiat. En même temps, elle montre, sur un mur, l'ombre « chaude » d'un être humain stationnant suffisamment longtemps près du mur pour élever sa température. Par comparaison avec les deux représentations précédentes, le robot identifie l'ombre et reconnaît le yucca.



Compteur

Nous ajoutons deux affichages à sept segments à notre port utilisateur qui nous permettront d'afficher continuellement le contenu des données du port utilisateur en hexadécimal.

Pour afficher des chiffres hexadécimaux, nous avons besoin de quatre bits (4 bits nous donnent 16 permutations de zéros et de uns). Ainsi, tout nombre binaire à 8 bits peut être représenté à l'aide de deux chiffres hexadécimaux : l'un pour les 4 bits inférieurs et l'autre pour les 4 bits supérieurs. Bien que chaque affichage soit composé de sept segments DEL, les diverses combinaisons peuvent être commandées par quatre lignes d'entrée si un décodeur logique est intégré dans le circuit. Les décodeurs sont des circuits qui traduisent des instructions provenant de l'ordinateur en signaux électriques pour ses périphériques, et vice versa. Pour cet exercice, nous pouvons acheter un circuit déjà construit. Il s'agit de la puce 7447 apparaissant dans la liste des pièces.

Le décodeur accepte pour chaque affichage quatre lignes d'entrée provenant du port utilisateur et fournit sept sorties via une séquence de portes logiques. Le circuit logique a été construit de telle façon que si les quatre lignes d'entrée étaient 0111, les segments appropriés s'allumeraient pour afficher le nombre 7 (0111 en binaire équivaut à 7 en hexadécimal).

Les chiffres hexadécimaux plus grands que neuf sont généralement représentés par les six premières lettres de l'alphabet (A à F). Vous noterez que la puce décodeur que nous utilisons a de curieuses configurations pour représenter ces chiffres. Il est probable que ces configurations peuvent être générées à l'aide des circuits logiques requis pour les chiffres zéro à neuf. Une optique de décodage supplémentaire aurait été nécessaire pour afficher les six derniers chiffres hex. Ainsi, en négligeant cette logique supplémentaire et en utilisant différents symboles pour représenter ces chiffres, le nombre de portes logiques dans le décodeur a été réduit.

Dès que le circuit d'affichage a été construit, nous pouvons afficher continuellement en hexadécimal le contenu du registre de données du port utilisateur, en utilisant les huit lignes de données fournies. Afin que chaque affichage puisse donner une information différente au même moment, une technique nommée « multiplexage » est employée. Essentiellement, les lignes d'entrée provenant du décodeur d'affichage passent d'un affichage à l'autre, les données présentes sur les lignes étant elles aussi changées. Si ce processus est effectué assez rapidement, tous les affichages multiplexés sont illuminés de façon continue, affichant les données présentes au moment où ils étaient connectés aux lignes de données.

Liste des pièces

N°	Article	N° Maplin
14	Résistances 330 ohms 0,4 watt	M330R
2	Décodeurs BCD 7447 à 7 segments	QX55K
1	Affichage à deux chiffres	BY66W
2	Prises de puce DIL à 16 broches	BL19V
1	Prise à angle droit 12 voies	YW30H
1	Fiche à angle droit 10 voies	YW19V
	Câble-ruban 8 voies*	
	Câble-ruban 7 voies*	
	Fil nu étamé*	
1	Carte de montage 50 trous par 36 bandes	FL09K
1	Boîtier en plastique 116 x 61 x 36 mm	LH60Q

* Il se peut que ces pièces soient des surplus des exercices précédents. La prise 12 voies n'est nécessaire que si vous désirez étendre le bus système. Celle-là et plusieurs liaisons de fils peuvent être omises sur la carte.

Nous pouvons illustrer le principe du multiplexage en utilisant les deux affichages à sept segments que nous construisons. Puisque le décodeur d'affichage représente 15 décimal par un blanc, nous pouvons utiliser ce nombre pour effacer un affichage en allumant l'autre. Le programme suivant, lorsqu'il est exécuté, sollicite un chiffre à afficher et semble présenter le chiffre simultanément sur les deux affichages. Cependant, une routine insère une temporisation qui ralentit l'oscillation entre les deux affichages. La temporisation est insérée lorsque la barre d'espacement est pressée. Nous pouvons voir, lorsque nous appuyons sur la barre d'espacement, que le chiffre passe en fait d'un affichage à l'autre. Lorsque nous relâchons la barre d'espacement, la temporisation est supprimée et l'oscillation est accélérée. Le chiffre semble de nouveau apparaître simultanément sur les deux affichages.

```

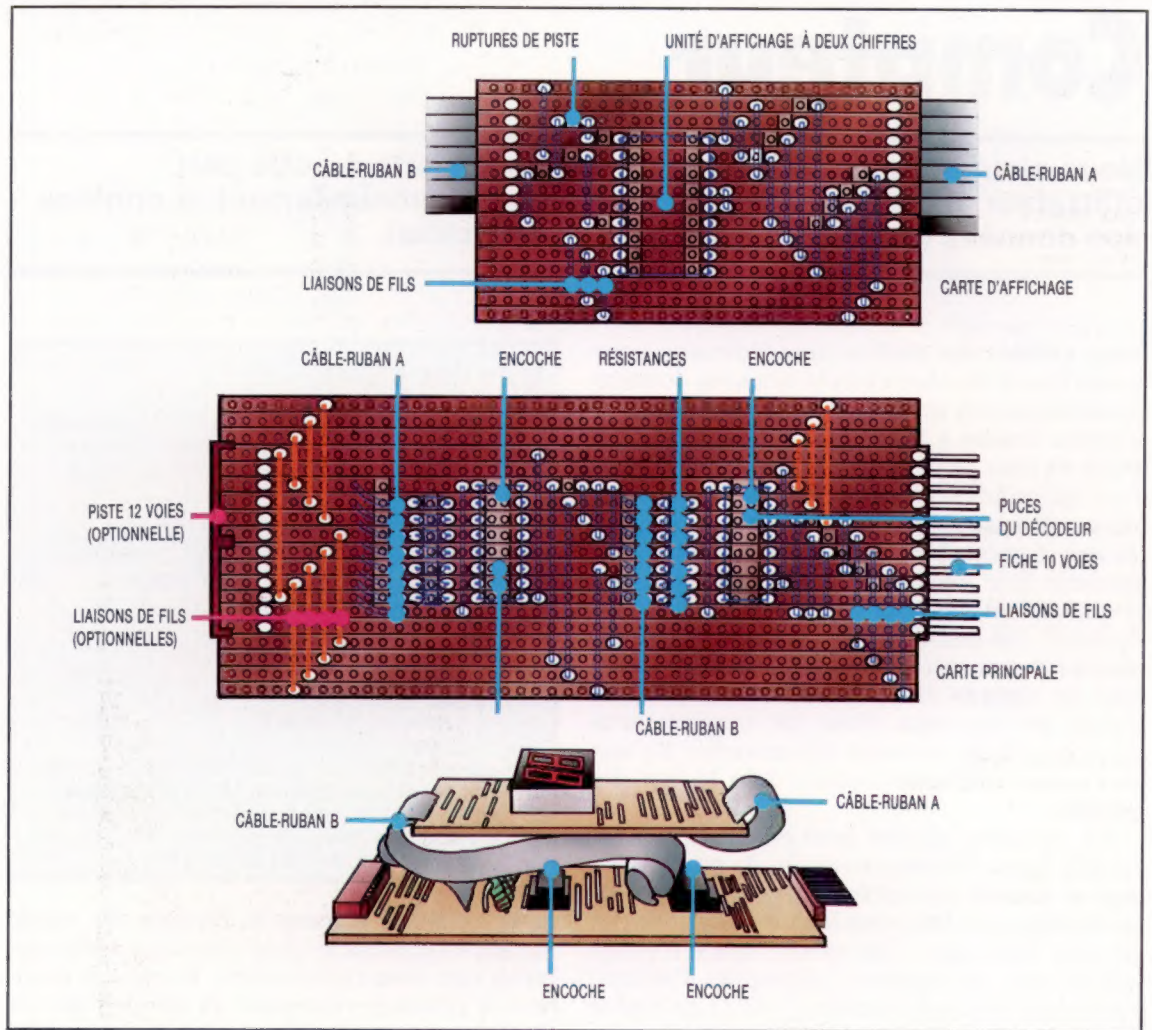
10 REM MULTIPLEXAGE BBC
20 DDR=$FEG2:DATREG=$FEG0
30 ?DDR=255
40 blanc_sauche=15+16
50 blanc_droit=15
55 :
60 REPEAT
70 INPUT "DONNEES A MULTIPLEXER" :données
80 ?DATREG=données+blanc_sauche
90 PROCient
100 ?DATREG=données+16+blanc_droit
110 PROCient
120 GOTD00
130 END
140 :
150 DEF PROCient
155 REM BARRE ESPACEMENT PRESSEE?

```



Montage

Coupez la carte de montage aux deux dimensions requises (19 pistes de 46 trous; 15 pistes de 28 trous). Coupez d'abord les ruptures de pistes sur les deux cartes. Soudez les deux prises de puces, puis les liaisons de fils et les résistances. Si la prise d'extension de bus n'est pas requise, omettez les liaisons rouges de l'illustration. Placez la prise 12 voies (optionnelle) et la fiche 10 voies sur la carte principale, et soudez l'unité d'affichage — les points étant placés vers l'extrémité de la carte. Soudez les câbles-rubans de façon qu'ils soient acheminés directement d'une carte à l'autre sans torsion. Enfichez les puces — assurez-vous qu'elles sont orientées comme sur l'illustration.



```

160 IF INKEY(-99)=-1 THEN PROCtemporis
170 ENDPROC
180 :
190 DEF PROCtemporis
200 FOR I=1 TO 500:NEXT
210 ENDPROC

```

```

10 REM MULTIPLEXAGE CBM 64
30 DDR=56579:DATREG=56577
40 POKEDDR, 255
50 LB=15+16*RB=15
60 INPUT "DONNEES A MULTIPLEXER":DT
70 POKEDATREG, DT+LB
80 GOSUB1000:REM PLUS LENT
90 POKE DATREG, DT*16+RB
100 GOSUB1000:REM PLUS LENT
110 GOT070
120 :
1000 REM SP PLUS LENT
1010 GETA#
1020 IFA#=" "THENGOSUB2000:REM TEMPORISATION
1030 RETURN
1999 :
2000 REM SP TEMPORISATION
2010 FORI=1 TO 250:NEXT
2020 RETURN

```

Une application consisterait à utiliser ces deux affichages à sept segments comme compteur hexadécimal. Ce dispositif affiche un compte du nombre d'impulsions entrées dans le port utilisateur à partir d'un interrupteur connecté à l'une des lignes du port. Cela peut sembler facile, jusqu'à ce que vous vous rendiez compte que les

huit lignes sont utilisées pour l'affichage. Si nous retenons l'une des lignes pour l'entrée, disons la ligne 0, le système d'E/S maintiendra toujours cette ligne au niveau haut. Si 128 (1000000) devait être placé dans le registre de données, ce nombre deviendrait instantanément 129 (1000001), parce que la ligne 0 est maintenue au niveau haut pour l'entrée. La solution consisterait à employer une technique similaire au multiplexage. Si nous prenons la ligne 0 pour accepter une entrée pendant un court moment et retenons toutes les lignes pour la sortie pendant une période plus longue, les affichages sembleront produire continuellement la valeur correcte, avec un scintillement de la valeur incorrecte provoqué par l'entrée sur la ligne 0.

```

10 REM COMPTEUR BBC
20 DDR=&FES2:DATREG=&FES0
50 compte=0
55 :
60 REPEAT
70 PROCentrée
72 PROCajouter
73 FORI=1TO40
75 PROCaffichage
77 NEXT I
80 UNTIL compte>255
90 END
999 :
1000 DEF PROCajouter
1010 IF drapreau=1 THEN compte=compte+1
1050 ENDPROC

```




```

1499 :
1500 DEF PROCentrée
1510 ?DDR=254
1515 drapeau=0
1520 IF(?DATREG AND 1)=0 THEN drapeau=1
1525 REPEAT UNTIL (?DATREG AND 1)=1
1530 ENDPROC
1999 :
2000 DEF PROCaffichage
2010 ?DDR=255
2030 ?DATREG=compte
2040 ENDPROC

```

```

10 REM COMPTEUR CBM 64
30 DDR,56579:DATREG=56577
40 CC=0:REM INIT COMPTE
50 :
60 GOSUB1000:REM ENTREE
70 GOSUB2000:REM AJOUTER
80 FOR I=1TO20
90 GOSUB3000:REM AFFICHAGE
100 NEXT I
110 IF CC 255 THEN60
120 END
999 :
1000 REM SP ENTREE
1010 POKEDDR,254
1020 FL=0
1030 IF(PEEK(DATREG)AND 1)=0 THEN FL=1
1040 IF(PEEK(DATREG)AND 1)<>1 THEN 1040
1050 RETURN
1999 :
2000 REM SP AJOUTER
2010 IF FL=1 THEN CC=CC+1
2020 RETURN
2999 :
3000 REM SP AFFICHAGE
3010 POKEDDR=255
3020 POKEDATREG,CC
3030 RETURN

```

Dans chaque cycle du programme, une boucle FOR...NEXT sert à répéter plusieurs fois la routine de mise en sortie de toutes les lignes lors de chaque routine de mise en entrée de la ligne 0. Pour la version du Commodore 64, vingt routines d'affichage sont exécutées pour chaque routine d'entrée. Ce rapport est porté à quarante pour la version du BBC en raison de la vitesse d'exécution supérieure du BBC Micro.

Avec ces valeurs, un scintillement est toujours perceptible, mais si le rapport est augmenté pour produire un affichage plus constant, il est possible de réduire suffisamment le temps d'entrée pour rater certaines entrées.



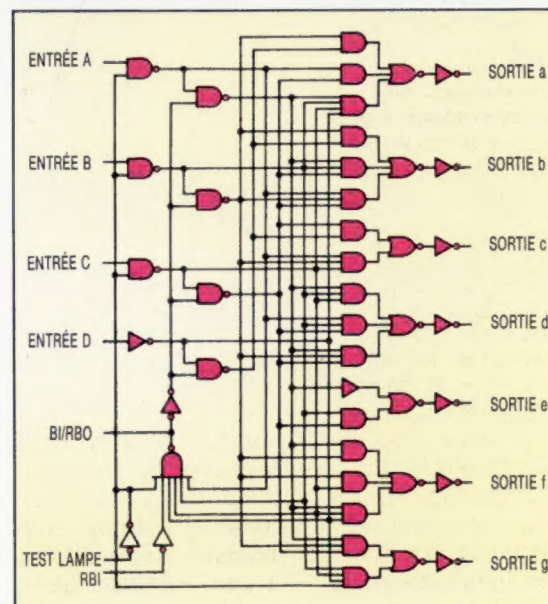
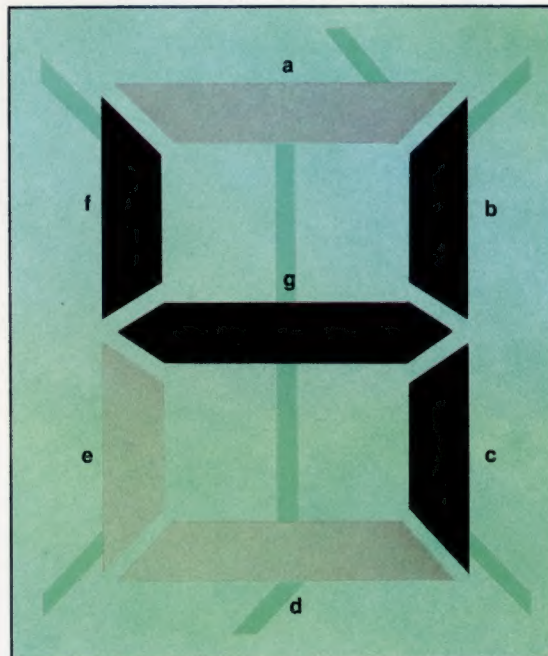
Boîtier terminé

Puisque chaque affichage numérique a besoin d'un code d'entrée à 4 bits, deux affichages peuvent être commandés à partir du port utilisateur. Ils sont placés dans une unité compatible avec les interfaces et les dispositifs que nous avons construits précédemment.

Décimal	Binaire	Sortie	Affichage
	D3D4D1D0	abcdefg	
0	0 0 0 0	0000001	0
1	0 0 0 1	1001111	1
2	0 0 1 0	0010010	2
3	0 0 1 0	0000110	3
4	0 1 0 0	1001100	4
5	0 1 0 1	0100100	5
6	0 1 1 0	1100000	6
7	0 1 1 1	0001111	7
8	1 0 0 0	0000000	8
9	1 0 0 1	0001100	9
10	1 0 1 0	1110010	
11	1 0 1 1	1100110	
12	1 1 0 0	1011100	
13	1 1 0 1	0110100	
14	1 1 1 0	1110000	
15	1 1 1 1	1111111	

Résultat

L'entrée vers l'affichage numérique provenant du port utilisateur est un nombre binaire à 4 bits. Un nombre à 7 bits est associé à chacun des nombres 0000 à 1111, chacun de ces 7 bits signale l'état de l'un des 7 segments de l'affichage. Dans ce code d'affichage, un bit zéro signifie que le segment correspondant doit être allumé et un 1 indique un segment éteint.

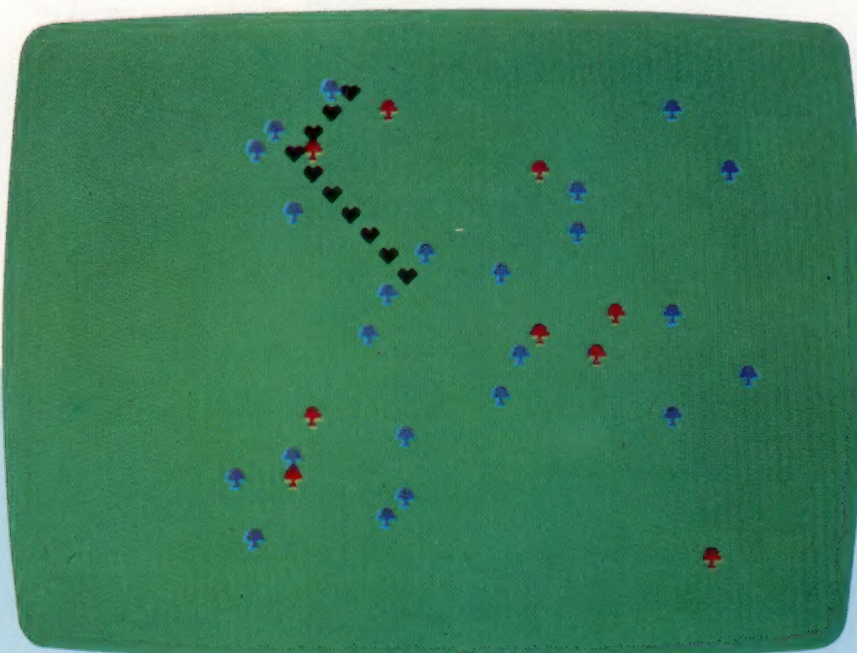


Décodeur/driver BCD 7447A à 7 segments

Les circuits internes de la puce illustrent la simplicité de sa logique — l'entrée à 4 bits est décodée en sorties 7 segments par les portes logiques. L'entrée de test allume tous les segments simultanément pour tester la puce.

Serpent sur MO5

Ce programme de jeu écrit en BASIC pour le Thomson MO5, par Pierre Monsaut, doit être tapé tel quel. Pour l'enregistrement sur cassette, utilisez l'instruction SAVE «SERPENT».



Dans ce jeu, vous êtes un serpent qui se déplace en ondulant sur l'écran. Le changement de direction s'effectue en tapant n'importe quelle touche. Pour pouvoir vous déplacer, vous devez vous nourrir. Heureusement, vous êtes entouré par un grand nombre de champignons. Mais attention, si les bleus sont excellents, vous devez absolument éviter les rouges : ils sont vénéneux. Chaque champignon bleu vous apporte suffisamment de calories pour avancer de dix lignes. Essayez de ne pas mourir de faim sans pour autant finir empoisonné !

```

10 REM *****
20 REM * SERPENT *
30 REM *****
40 CLEAR ,,2
50 GOSUB 540
60 D$=INKEY$
70 IF D$<>" " THEN D=-D
80 X=X+D
90 IF X<0 THEN X=1
100 IF X>39 THEN X=38
110 IF POINT(X*8+4,Y*8+4)=1 THEN 260
120 IF POINT(X*8+4,Y*8+4)=4 THEN S=S+10:
H=H+10:BEEP
130 LOCATE X,Y
140 COLOR 0
150 PRINT S$;
160 LOCATE 0,24,0
170 PRINT
180 LOCATE INT(RND*38)+1,24
190 COLOR 4
200 PRINT C$;
210 IF RND>0.5 THEN LOCATE INT(RND*38)+1
,24:COLOR 1:PRINT C$;
220 S=S-1
230 IF S=0 THEN 260
240 H=H+1
250 GOTO 60
260 BEEP
270 LOCATE 0,24
280 PRINT
290 LOCATE X,Y-1
300 COLOR 5
310 PRINT S$;
320 FOR I=1 TO 5
330 BEEP

```

```

340 FOR J=1 TO 50
350 NEXT J
360 NEXT I
370 IF INKEY$<>" " THEN 370
380 IF H>R THEN R=H
390 LOCATE 4,20
400 COLOR 0
410 PRINT "SCORE :";H;
420 LOCATE 23,20
430 PRINT "RECORD :";R;
440 LOCATE 15,23
450 PRINT "UNE AUTRE ?"
460 D$=INKEY$
470 IF D$="" THEN 460
480 IF D$<>"N" THEN 520
490 SCREEN 4,6,6
500 CLS
510 END
520 GOSUB 610
530 GOTO 60
540 CLS
550 SCREEN 2,2,2
560 DEFINT A-Z
570 DEFGR$(0)=0,0,102,255,219,126,60,24
580 DEFGR$(1)=60,126,126,255,255,24,24,6
0
590 S$=GR$(0)
600 C$=GR$(1)
610 CLS
620 S=100
630 H=0
640 D=1
650 X=19
660 Y=10
670 RETURN

```



Du micro à la compo (suite)

La photocomposition bouleverse bien des traditions dans le monde de l'imprimerie. La galaxie Gutenberg évolue vers celle de McLuhan. L'image remplace les lettres de plomb.

Nous avons vu précédemment ce que la photocomposition signifiait dans la pratique. Ainsi, la création de textes, de documents sur papier, connaît des transformations profondes; fini, les lettres de plomb posées une à une à la main pour former des lignes.

Si les imprimeries modernes font appel à des machines importantes, la photocomposition peut très bien s'accommoder de micro-ordinateurs. Étudions à présent deux exemples typiques de logiciels: le premier, Cosy 200, est associé à un ordinateur Siemens; le second, PagePlanner, au contraire, fonctionne avec des micro-ordinateurs IBM PC ou d'autres, compatibles.

Le système Cosy 200

Siemens propose un progiciel de composition particulièrement adapté aux journaux et périodiques, Cosy 200. Il fonctionne sur la gamme d'ordinateurs 7500/7700 de ce constructeur. La saisie des textes se fait sur terminal 9750-62. La photocomposition est effectuée par Digiset 40T30.

Ce système permet la mise en forme typographique des textes. Il comporte des programmes spécifiques pour les besoins de la presse quotidienne et périodique. Des programmes rédactionnels permettent, notamment, de mémoriser des tableaux, gérer des photos, générer des filets et des cadres, et assurent un certain nombre de fonctions de gestion: celle des messages d'agences de presse, ainsi que des annonces, incluant le traitement des logos, la saisie et la facturation en temps réel des annonces.

Avec le système Cosy 200, les centres de production peuvent être complètement décentralisés, écrans et photocomposeuses étant reliés au site central par des lignes téléphoniques louées. C'est le cas du *Progrès de Lyon*: le site central est à Lyon, les ordinateurs Siemens et deux photocomposeuses sont en service à Saint-Étienne (*l'Espoir, la Tribune*) et une à Louhans (*l'Indépendant*).

Ce système a été conçu pour répondre aux besoins de puissance et d'urgence des quotidiens et s'adapte à l'organisation des journaux.

PagePlanner

Le logiciel PagePlanner se compose de trois modules intégrés: la saisie et la correction de textes, la justification et les coupures, et enfin, la mise en page. Il fonctionne non pas sur un gros ordinateur, comme le système précédent, mais sur micro: IBM PC ou les compatibles, munis des



Arch. Ed. Atlas

systèmes d'exploitation standard MS-DOS et CP/M. La configuration du micro doit comporter au minimum 128 K de mémoire et une carte graphique pour la visualisation du montage de page. L'architecture du système peut se construire autour de concentrateurs ou de réseaux locaux.

Toutes les fonctions de PagePlanner sont accessibles par des menus, avec l'assistance sous forme de suggestions pour les différentes phases de l'édition, de la composition et de la mise en page.

PagePlanner permet aux utilisateurs de traiter leurs textes en partant d'une saisie sous un logiciel de traitement de texte standard, avant d'aboutir à l'éditeur typographique de PagePlanner.

Fini le temps où l'on composait en plomb, la photocomposition et ses bromures ont pris la relève.

Revendeurs de PagePlanner

Pour la France

Compaudit : 13, allée Renoir
77144 Chalifert
Tél. : (64) 36.87.63.

Pour la Belgique

MicroCompo : Av. Victor-Olivierlaan, 19-21
1070 Bruxelles



Menus PagePlanner

MENU PRINCIPAL

- 1. TEXTE SAISIE/CORRECTION
- 3. JUSTIFICATION
- 5. MISE EN PAGE
- 7. PHOTOCOMPOSITION
- 9. IMPRIMANTE
- 10. COMMUNICATIONS
- 11. GESTION DISQUE
- 12. UTILITAIRES
- 14. RETOUR SYSTEME EXPLOITATION

TEXTE SAISIE/CORRECTION

- 1. Vérifier espace disque disponible
- 2. Changement disque des données (lecteur B)
- 3. Répertoire des fichiers
- 4. Création d'un fichier
- 5. Edition d'un fichier existant
- X. Retour au menu principal

Sélectionner l'option

JUSTIFICATION ET COUPURES

- 1. Vérifier espace disque disponible
- 2. Changement disque des données (lecteurs)
- 3. Répertoire des fichiers
- 4. Justification d'un fichier
- 5. Coupures avant préfixe? (active) « oui »
- 6. Coupures avant suffixe? (active) « oui »
- 7. Coupures logiques? (active) « oui »
- X. Retour au menu principal

Sélectionner l'option

MISE EN PAGE

- 1. Création fichier page
- 2. Extension d'une page existante
- 3. Exécution d'une page
- 4. Changement disque des données (lecteur B)
- 5. Répertoire des fichiers
- X. Retour au menu principal

La saisie des textes se fait maintenant au clavier. Les « Linotype », après de bons et loyaux services ont pris le chemin des musées de la typographie.

A partir de cet éditeur spécialisé, les textes peuvent être composés avec les caractères préalablement choisis et incorporés dans le micro.

L'étape suivante est la mise en page, avec la spécification du gabarit en hauteur et en largeur, suivie de l'appel des textes à incorporer dans la page. On définit ensuite des colonnes, des filets et des « boîtes », ainsi que des zones protégées pour l'incorporation de tableaux, schémas, similes ou autres documents fournis. Les textes appelés seront automatiquement « coulés » dans la maquette d'après les directives précisées. La page en cours d'exécution peut à tout moment être abandonnée pour être reprise ultérieurement. Une fois complétée, elle sera stockée avec toutes les instructions de montage et pourra sortir vers la photocomposeuse.

Plusieurs micros reliés en réseau peuvent envoyer des pages en direction d'une imprimante ou photocomposeuse commune. Les fichiers textes ou pages peuvent aussi transiter par le réseau pour l'envoi sur un autre poste, par exemple celui d'un superviseur ou coordinateur.



Renée Panier/Ed. Atlas



Qui fait quoi?

Nous citons ici un certain nombre de sociétés effectuant de la photocomposition à partir de traitement de texte. Quelques précisions sont fournies sur les services proposés par ces firmes, ainsi que leurs adresses.

Buromatic Royer : société de services qui a étendu ses activités à la photocomposition, elle émet à partir de systèmes Wang en direction de photocomposeuses Wang, Siemens, Berthold et Linotype.

9, rue Professeur-Florence, 69003 Lyon.
Tél. : (7) 223.42.70

C.I.E.R. : il s'agit d'une société de conseil qui a mis au point un logiciel permettant de photocomposer à distance, à partir d'un matériel Canon. 2, rue Moncey, 75009 Paris.
Tél. : (1) 874.88.98.

Compogram : les textes sont transmis sur photocomposeuse Compugraphic System MCS par l'intermédiaire d'un modem Racal Milgo MPS 1222. 174, quai de Jemmapes, 75010 Paris.
Tél. : (1) 238.22.65.

Dawart : la firme met des IBM PC à la disposition de ses clients et est équipée d'un transcodeur Seat 201 et d'une Linotron 202. Elle accepte les disquettes au format 3740/41 ainsi que des disquettes IBM PC ou compatibles.

13, rue des Archebusiers, 75003 Paris.
Tél. : (1) 887.44.40.

Diagramme : cette firme, qui effectue elle-même l'enrichissement des textes, peut lire des disquettes 8 pouces au format 3740, ou recevoir, grâce à un modem SAT, du texte photocomposé sur Linotron 202. 48, bd des Batignolles, 75017 Paris.
Tél. : (1) 293.11.53.

Digi France : elle peut recevoir des textes par téléphone ou lire des disquettes au format IBM 3740 et TTX. 69, rue Chabrol, 75010 Paris.
Tél. : (1) 246.53.88.

Écriture Service : entreprise de travail à façon, qui a développé un service de photocomposition en liaison avec la société Coupé. Elle assure la lecture de disquettes 8 pouces compatibles IBM 3740 et de bandes magnétiques 1600 bpi, et peut recevoir sur réseau commuté à partir de tous matériels équipés d'une interface (carte et logiciel) de réception et d'envoi de données par modem. Le matériel utilisé à la réception est un mini-ordinateur VAX 730 et des photocomposeuses Linotron 202. 8, rue de la Mitrie, 44000 Nantes.
Tél. : (40) 49.24.40.

Gedev : cette firme de services est particulièrement axée sur la traduction de documents et travaille sur des matériels TTX80 et Visiotexte. 34 bis, rue Vignon, 75009 Paris.
Tél. : (1) 742.84.14.

HTI : 2, rue Damiette, 75002 Paris.
Tél. : (1) 233.29.83.

Ika : société de services en informatique, Ika peut réaliser des travaux sur imprimante à laser Rank Xerox 9700 et Burroughs 9370, ou sur photocomposeuse Digiset 40T2. Les textes peuvent lui être transmis sur bandes magnétiques ou disquettes IBM 34 et 38. Ika accepte aussi les disquettes des matériels TTX, Système 6 et Visiotexte d'IBM, Rank Xerox 860 et Wang. 112, bd Haussmann, 75008 Paris.
Tél. : (1) 522.13.69.

Imprimerie Blanchard : cet imprimeur dispose d'un transcodeur Seat et d'un système de traitement Mopas. 6, av. Descartes, 1, 92350 Le Plessis-Robinson.
Tél. : (1) 630.22.72.

Imprimerie Royer : elle effectue l'enrichissement de textes saisis sur matériel Wang, IBM, Alcatel et Bull, transmis par liaison téléphonique, et reçoit sur transcodeur Seat et système de photocomposition CR Tronic (Linotype). B.P. 1142, 78204 Mantes-la-Jolie Cedex.
Tél. : (3) 092.72.88.

Industrial Computing : spécialisée dans le matériel Dec, cette société peut lire des disquettes Dec ou compatibles et recevoir à partir de tous matériels Dec et compatibles VT 100. Des claviers VT 100, VT200, PC300 et Datakey peuvent être mis à la disposition des clients. 72, quai Carrières, 94220 Charenton.
Tél. : (1) 893.99.00.

Iota : cette firme effectue la lecture de disquettes Rank Xerox 860 et MDS Série 21 et peut recevoir des textes en provenance de tout matériel possédant le protocole standard IBM, BSC 2780. Les textes sont traités sur photocomposeuse Digiset Siemens. 34, av. du Roule, 92200 Neuilly-sur-Seine.
Tél. : (1) 722.48.50.

Maury : Imprimeur. ZI, 45330 Malesherbes.
Tél. : (38) 34.60.25.

MicroCompo : firme belge fabriquant et vendant du matériel de photocomposition, MicroCompo a développé un éditeur de texte qui permet une saisie simplifiée. La console CompoType est mise en dépôt ou en location chez les clients. Av. Victor-Olivierlaan, 19-21, 1070 Bruxelles.
Tél. : 2/254.00.52.
(Revendeur de PagePlanner.)

Multicompo : elle peut lire des disquettes Visiotexte ou recevoir par télétransmission à partir de matériel IBM (PC et Visiotexte) et Philips 5020. En sortie, elle dispose de matériel Volt Autologic. 8, rue du Marché-Popincourt, 75011 Paris.
Tél. : (1) 700.54.14.

Nord Compo : cet imprimeur du nord de la France assure la lecture de bandes magnétiques et peut recevoir par réseau commuté, à partir de tous les matériels utilisant des disquettes. Les documents sont reçus sur système Miles 300 et MCS Compugraphic. 7, rue de Fives, B.P. 123, 59650 Villeneuve-d'Asq.
Tél. : (20) 91.01.32.

Secrétaire Bureautique 34 : cette société travaille par télétransmission avec des imprimeries équipées de matériels Compugraphic et AM International, mais est prête à offrir ses services à d'autres marques. La saisie se fait sur système de traitement de texte Olivetti ETS 1010 et SMH Alcatel. 4, rue de la République, 34000 Montpellier.
Tél. : (67) 58.30.68.

Secrétaire Bureautique 38 : cousine germaine de la Secrétaire Bureautique 34, elle a établi une liaison téléphonique entre matériel Olivetti ETS 1010, Compugraphic 7600 et AM International. Route de Grenoble, 38470 Vinay.
Tél. : (76) 36.72.52.

Sofiac : firme dotée d'un important parc d'ordinateurs, de systèmes de photocomposition et de moyens d'entrée divers (notamment lecteurs optiques dont un Kurzweil), elle peut lire les bandes magnétiques et des disquettes en permanence de nombreux matériels : IBM PC, Apple II, Rank Xerox 820, Osborne, Kaypro, Wang, et les disquettes 8 pouces sous CP/M. 8, rue de Furstenberg, 75006 Paris.
Tél. : (1) 634.20.20.

Speco & Co : cette société a mis au point des liaisons avec des systèmes Goupil et Visiotexte, et utilise une photocomposeuse Berthold. 9, avenue Leclerc, 69007 Lyon.
Tél. : (7) 872.00.06.

S3P : spécialisée dans le matériel Wang, cette firme peut recevoir des disquettes 8 pouces Wang ou par télétransmission en provenance de Philips, IBM, Rank Xerox et SMH Alcatel. S3P dispose d'une photocomposeuse Wang 5548Z. 28, rue Caumartin, 75009 Paris.
Tél. : (1) 742.11.50.

Télécompo : filiale de l'imprimerie Rosay, Télécompo peut recevoir des disquettes en provenance de Victor S1, NBI, IBM 3740 et Exidy système, ainsi que des bandes magnétiques à 1600 bpi. Par télétransmission, elle travaille avec de nombreux systèmes. Olivetti, SMH Alcatel, Bull TTX, Philips, Dec, Apple et Commodore. Pour la composition, elle est équipée d'un système Cerci. 13-15, av. du Petit-Parc, 94300 Vincennes.
Tél. : (1) 328.18.63.



Machine à calculer

Tout le monde peut profiter d'un progiciel tableur et générateur de modèles financiers. Ces articles sont conçus comme un guide d'utilisation sur micro des tableurs résidant sur cassette.

Il existe plusieurs progiciels de modèles financiers pour des micro-ordinateurs très répandus comme le Spectrum de Sinclair, le Commodore 64 et le Modèle B de BBC. Cet ensemble d'articles est destiné à vous procurer un guide pratique et progressif d'utilisation des tableurs dans des applications quotidiennes et variées.

Le cœur de tout tableur est un tableau électronique, divisé en colonnes et en lignes, semblable à une grande feuille de papier millimétré destinée aux graphiques ou aux modèles financiers. L'écran de télévision, le moniteur, joue le rôle d'une fenêtre suppressible pouvant afficher toute partie du tableau (le tableau est bien entendu plus grand que les quatre ou cinq colonnes et les dix ou quinze lignes affichables).

L'intersection d'une colonne et d'une ligne s'appelle « case ». Une case peut contenir un nombre, du texte ou une formule mathématique. Le tableur utilise un curseur, habituellement un bloc à l'écran en double-brillance, pouvant être déplacé par les touches du curseur. Toute saisie de données est supposée concerner la case occupée couramment par le curseur.

Voici, résumée, la conception de base d'un tableur. Dans cet article, nous nous consacrerons à un progiciel appelé Vu-Calc distribué par Psion et disponible pour le Spectrum et le Modèle B. Nous étudions ici sa version BBC qui est pratiquement identique à celle du Spectrum (à l'exception de quelques différences contrariantes bien que mineures, dans le nom de certaines commandes).

Vu-Calc montre la flexibilité et la grande utilité d'un tableur. Il ne comporte pas les caractéristiques de modélisation financière de progiciels complexes tels que Lotus 1-2-3. Vous ne pouvez diviser l'écran verticalement ou horizontalement pour afficher simultanément plusieurs parties du document, et vous ne disposez pas non plus de toutes les caractéristiques sophistiquées conçues pour un usage professionnel. Mais vous pouvez utiliser Vu-Calc de manière profitable pour établir des modèles financiers très utiles que vous pouvez sauvegarder sur cassette.

La version BBC comporte un maximum de 28 colonnes numérotées à partir de 1, et 52 lignes dénommées selon l'alphabet, avec les lignes venant après « Z » dotées d'une double lettre reprenant l'ordre alphabétique : « AA », « BB », ... Cette capacité n'est pas très importante selon les standards des tableurs, mais la limite vient de la faible mémoire du BBC Micro, qui a seulement 32 K.

Établir un modèle

Une des applications les plus utiles pour un tableur sur micro-ordinateur domestique est le budget annuel de la famille, qui a également le mérite d'être un modèle financier relativement simple à établir.

Une fois construit, un tel modèle vous permet de voir en un coup d'œil l'impact de l'augmentation de tout poste de dépenses, une note téléphonique trop importante par exemple ou un



voyage imprévu devront figurer au poste « dépenses imprévues ».

En d'autres termes, une fois le modèle établi, vous pouvez expérimenter des modifications ponctuelles sur les données, afin d'apprécier les modifications d'ensemble en résultant.

Des avantages évidents

La première étape consiste à dresser par écrit la liste de tous les postes budgétaires. Ensuite, vous estimez la dépense mensuelle probable pour chacun. C'est là que les avantages du tableur deviennent évidents.

Les catégories de dépenses sont entrées dans la première colonne, les colonnes suivantes recevant les noms de mois, JAN, FEV, MAR, AVR, etc. Vu-Calc demande que tout texte saisi soit précédé de guillemets, mais cette ponctuation n'apparaît pas à l'affichage. Les premières colonnes et rangées de notre tableau se présenteront donc de la manière suivante :

	1	2	3	4
A		JAN	FEV	MAR
B	Voiture-crédit			
C	Emprunt	150		
D	Impôts			

Voici donc la structure de base de notre modèle. Il nous reste à entrer les valeurs appropriées, et nous utiliserons pour cela la commande REPLICATE (recopier) de Vu-Calc.

Vu-Calc possède un nombre limité de commandes : toutes sont préfixées du signe *. Lorsque vous tapez * sur une case vide, le programme passe en mode commande. REPLICATE est certainement la commande la plus utile, du fait que la partie la plus fastidieuse de la création d'un modèle financier est d'entrer toutes les valeurs à utiliser. C'est avant tout une commande qui fait gagner du temps.

Une partie classique d'un modèle de budget ménager est constituée des frais fixes, tels que les impôts, les emprunts ou le loyer. Si les emprunts sont par exemple de 150 F par mois, la commande REPLICATE placera cette somme dans les douze cases appropriées.

Lorsque vous appelez la commande REPLICATE en tapant « *R », un message d'aide apparaît en haut de l'écran, au-dessus du modèle lui-même, indiquant :

Replicate — Tapez la saisie pour la case à répéter, faites RETOUR-CHARIOT pour revenir à la case courante.

Ce message vous demande les cases à répéter. Vous remarquez que vous ne pouvez copier qu'une seule case et non tout un ensemble de cases, ce qui constitue une des limitations les plus manifestes de Vu-Calc. La case est indiquée par ses coordonnées, la lettre pour la ligne étant donnée en premier, suivie par le numéro de la colonne, par exemple C2. Après cela, le message vous demande :

Donnez la zone concernée par la duplication des données.

Une zone de positions pour Vu-Calc est définie en spécifiant la première case — la plus à gauche en haut, et la dernière, en bas à droite. Si vous pensez à une zone comme à un pavé ou à une boîte, vous devez spécifier la position supérieure gauche et la position inférieure droite.

Dans notre exemple, nous voulons demander au programme de placer 150 F dans la zone des positions comprises entre C3 et C13, sur la ligne C, des rangées 3 à 13, c'est-à-dire des mois de février à décembre. Le format, à cette fin, est *R, C2,C3-C13. Cela remplit instantanément les positions voulues. La vraie puissance de REPLICATE est cependant de recopier des formules mathématiques d'une position à l'autre.

Nous poursuivons avec les autres postes budgétaires de la même manière. Notez que si vous décidez qu'une valeur pour un mois déterminé doit être plus grande ou plus petite que la valeur standard, vous pouvez toujours déplacer le curseur sur la case appropriée pour saisir la nouvelle valeur. Cela a pour effet d'effacer immédiatement en écriture l'ancienne valeur.

Dans notre modèle, la colonne 14 représentera la colonne des totaux. Il y aurait peu d'intérêt à utiliser un tableur s'il fallait par ailleurs utiliser un calculateur pour effectuer les totaux récapitulatifs de chaque ligne, la somme des valeurs pour tous les mois par poste. Vu-Calc peut donc être utilisé pour effectuer les totaux par ligne et par colonne. La commande \diamond indique que vous désirez effectuer le total des valeurs que comporte une zone de positions. La zone est spécifiée de la même manière que précédemment. Aussi, pour faire le total des emprunts sur l'année et mettre le résultat dans la case C14, il vous suffit de placer le curseur en C14 et de taper : \diamond C2:C13. Le résultat — 1800 ici — est immédiatement affiché.

Nous aurions pu tout aussi bien mettre une formule mathématique dans la case 14. Au lieu d'ajouter toutes les valeurs nous aurions pu faire C2*12. Vu-Calc aurait considéré cela comme une formule du fait que C n'est pas précédé de guillemets. Le résultat aurait été immédiat : 1800. Cela montre qu'il y a souvent plus d'une manière de parvenir à un résultat donné.

Tableur sur plusieurs micros

Commodore 64

Busicalc : cassettes/disquettes, Supersoft, Winchester House, Canning Road, Harrow HA3 7SJ.

Insta-Calc Graphic : cartouches/disquettes; Dataview Wordcraft Ltd., Radix House, East Street, Colchester CO1 2XB.

BBC Micro

Vu-Calc : cassettes, Psion Ltd., 2 Huntsworth Mews, Gloucester Place, London NW1 16DD.

Spectrum

Vu-Calc : cassettes, Psion Ltd.

Massacreur de dragons

Nous continuons notre étude des figures graphiques sous LOGO, par l'algorithme de la « courbe des poursuites », le problème des trois bugs et une stratégie complexe d'interception.

Nous donnons ici les procédures pour un jeu qui utilise les figures graphiques LOGO. Vous pilotez un dragon qui essaye d'atteindre une cité pour la détruire.

La défense de la cité est assurée par un chevalier volant (que dirige l'ordinateur) qui s'efforcera de tuer le dragon. Vous pilotez le dragon par une manette de jeu. Si vous parvenez à échapper au chevalier et à vous approcher suffisamment de la cité, le souffle du dragon mettra la ville en feu.

Pour exécuter le jeu, vous devez charger le fichier FIGURES, définir vos formes géométriques, saisir les procédures et taper ensuite JEU. Une fois l'initialisation effectuée, la procédure JEU appelle JOUER, qui est la procédure centrale. JOUER fait déplacer le dragon et le chevalier tour à tour,

et vérifie si le dragon a atteint la ville ou si le chevalier a touché le dragon.

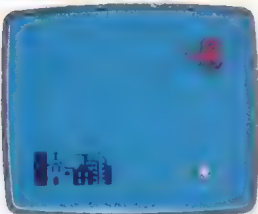
Les commandes de couleur disponibles sont très directes. Pour donner la couleur du fond, utiliser FOND suivi du numéro de couleur désiré. Pour donner la couleur d'une figure et de la ligne que cette figure trace lorsque son crayon est baissé, utilisez COULEURCRAYON. Les numéros des couleurs reçoivent un nom dans INIT.VARIABLES, et nous pouvons donc spécifier les couleurs par leur nom, en utilisant des commandes telles que COULEURCRAYON : ROUGE.

Dans la procédure JOUER, la ligne :

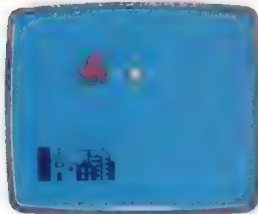
```
SI TOUCHE? ALORS DRAGON.DÉTRUIT
```

est générée afin de tester si le chevalier a touché le dragon. La procédure TOUCHE? est une bonne

Chevaliers et dragons



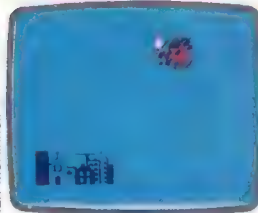
Le jeu en pleine action



La ville en flammes



Dragon vaincu



Chevalier contre dragon

POUR JEU

INIT.VARIABLES

METTRE.EN.PLACE.ÉCRAN

JOUER

FIN

POUR INIT.VARIABLES

FAIRE «FLAMME1

FAIRE «FLAMME2

FAIRE «DRAGON3

FAIRE «CHEVALIER4

FAIRE «VILLE5

FAIRE «VILLE6

FAIRE «ROUGE2

FAIRE «NOIR0

FAIRE «BLEU6

FAIRE «ORANGE8

FAIRE «JAUNE7

FIN

POUR METTRE.EN.PLACE.ÉCRAN

DESSINE

PLEINÉCRAN

FOND :BLEU

DIRE 0

LÈVEPLUME

CACHETORTUE

DIRE :FLAMME1

CACHETORTUE

CACHETORTUE

DIRE :FLAMME

CACHETORTUE

POSITION :DRAGON 100 100 :ROUGE

GRANDX GRANDY

POSITION :VILLE (- 52) (- 80) :NOIR

GRANDX GRANDY

POSITION :VILLE1 (- 100) (- 80) :NOIR

GRANDS GRANDY

DIRE4

LÈVEPLUME

POSITION :CHEVALIER (100) (- 100) :JAUNE

PETITX PETITY

FIN

POUR JOUER

DÉPLACEMENT.DRAGON

SI DISTANCE :DRAGON :VILLE < 50 ALORS

VILLE.DÉTRUITE STOP

SI DISTANCE :DRAGON :VILLE1 < 50 ALORS

VILLE.DÉTRUITE STOP

DÉPLACEMENT.CHEVALIER

SI TOUCHE? ALORS DRAGON.DÉTRUIT STOP

JOUER

FIN

POUR DÉPLACEMENT.DRAGON

DIRE :DRAGON

BOUGERMANETTE MANETTE1

AV 10

FIN

POUR BOUGERMANETTE :DIR

SI :DIR < 0 STOP

DONNERSENS :DIR ★ 45

FIN

POUR DISTANCE :A :B

DIRE :A

FAIRE «X1 XCOR

FAIRE «Y1 YCOR

DIRE :B

(PAGE 1175)

illustration de la manière dont nous pouvons écrire nos propres conditions de test avec LOGO. Elle donne en résultat la valeur «VRAI, ou la valeur «FAUX, utilisée ensuite en saisie aux instructions SI. Le résultat «VRAI suscitera ensuite l'exécution de l'action conditionnelle.

TOUCHÉ? utilise une procédure en provenance du fichier FIGURES, TOUCHERFIGURE?, qui donne la valeur «VRAI lorsqu'une figure graphique touche la figure courante. TOUCHÉ? affecte en premier la figure courante au dragon.

La commande MANETTE prend en saisie 0 ou 1 (ce qui correspond aux ports 1 et 2). Le résultat en sortie est - 1 lorsque la manette est en position centrale, 0 lorsqu'elle est vers le haut, 1 lorsqu'elle est à 45°, 2 à 90°, et ainsi de suite jusqu'à 7. Nous donnons ici simplement au dragon la direction 45° multipliée par le nombre en sortie.

Les explosions et les effets similaires sont facilement réalisables avec des figures graphiques.

L'ordinateur contrôle le chevalier, mais utilise une stratégie défensive simple; le chevalier se dirige droit sur le dragon. Dans l'état actuel du jeu, le dragon peut échapper aisément au chevalier.

Comment améliorer la stratégie défensive du chevalier? Une manière simple est d'accroître sa vitesse, et la faire passer de 10 à 11 rend l'échappée très difficile au dragon.

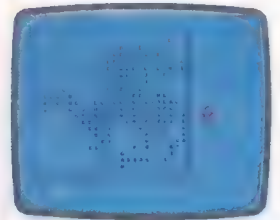
Projets des 3 punaises

Les punaises démarrent des trois angles d'un triangle :

```

POUR PUNAISES
  POSITIONNER DÉPLACER.PUNAISES
FIN
POUR POSITIONNER
  TRACER PLEINÉCRAN DIRE 0 CACHETORTUE
  LÈVEPLUME POSITIONNERXY (- 100)
  (- 100) TRI 200 POSITION 1 (- 100)
  (- 100) POSITION 2 0 73 POSITION 3 100
  (- 100)
FIN
POUR TRI :CÔTE
  POSEPLUME 3 [AV :CÔTE DR 120]
  LEVEPLUME
FIN
POUR POSITION :NO :X :Y
  DIRE :NO PLACERFORME3 LÈVEPLUME
  DETERMINERXY :X :Y POSEPLUME DÉBUT
FIN
POUR DÉPLACER.PUNAISES
  SUIVRE 1 2 SUIVRE 2 3 SUIVRE 3 1
  DÉPLACER.PUNAISES
FIN
POUR SUIVRE :A :B
  DIRE :B FAIRE «X XCOR FAIRE «Y YCOR DIRE
  :A DIRIGER VERS :X :Y AV 10
FIN
  
```

Éditeur de figures graphiques



```

FAIRE «X2 XCOR
FAIRE «Y2 YCOR
SORTIE RACINECARRÉE ((:X1- :X2) ★ (:X1- :X2) + (:Y1- :Y2)
★ (:Y1- :Y2))
  
```

FIN

POUR VILLE DÉTRUITE

```

DIRE :VILLE
FAIRE «X XCOR
FAIRE «Y YCOR
FLASH :X :Y :ORANGE
DIRE :VILLE1
FAIRE «X2 XCOR
FAIRE «Y2 YCOR
FLASH1 :X2 :Y2 :ORANGE
CACHER.FIGURE
ÉCRANEXPLOSION
RÉPÊTE 3 [AFFICHER »]
AFFICHER [VILLE DÉTRUITE]
  
```

FIN

POUR FLASH :X :Y :COULEUR

```

DIRE :FLAMME
COULEURTRAIT :COULEUR
POSITIONNERXY :X :Y
DÉBUT
RÉPÊTE 6 [PETITX PETITY ATTENDRE GRANDX GRANDY
ATTENDRE]
  
```

FIN

POUR FLASH1 :X2 :Y2 :COULEUR

```

DIRE :FLAMME1 COULEURTRAIT :COULEUR
POSITIONNERXY :X2 :Y2
DÉBUT RÉPÊTE 6 [PETITX PETITY ATTENDRE GRANDX
GRANDY ATTENDRE]
  
```

FIN

POUR DÉPLACEMENT.CHEVALIER

```

DIRE :DRAGON
FAIRE «X XCOR
FAIRE «Y YCOR
DIRE :CHEVALIER
DIRIGER VERS :X :Y
AV 10
  
```

FIN

POUR TOUCHÉ?

```

DIRE :DRAGON
SI FIGURETOUCHÉE? ALORS RÉSULTAT «VRAI
RÉSULTAT «FAUX
  
```

FIN

POUR DRAGON DÉTRUIT

```

DIRE :DRAGON
FAIRE «X XCOR
FAIRE «Y YCOR
FLASH :X :Y :NOIR
CACHER.FIGURE
ÉCRAN.EXPLOSION
RÉPÊTE 3 [AFFICHER »]
(AFFICHER [DRAGON TUE A UNE DISTANCE DE]
DISTANCE :DRAGON :VILLE)
VILLE
  
```

FIN

POUR CACHER.FIGURE

```

DIRE :FLAMME CACHETORTUE
DIRE :FLAMME1 CACHETORTUE
DIRE :VILLE CACHETORTUE
DIRE :VILLE1 CACHETORTUE
  
```

FIN

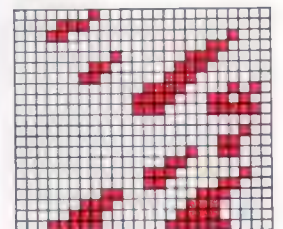
POUR FIN

```

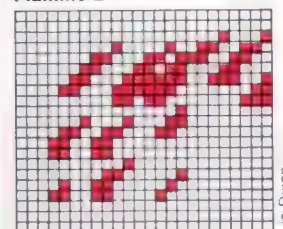
RÉPÊTE 100 []
  
```

FIN

Flamme 1



Flamme 2



La meilleure stratégie

La meilleure stratégie (pour les deux parties) est de se diriger vers la ville selon une direction perpendiculaire à la ligne reliant le dragon et le chevalier (comme l'indique la figure).

POUR DÉPLACEMENT.CHEVALIER
 DIRE :DRAGON FAIRE «DX XCOR FAIRE «DY
 YCOR DIRE :CHEVALIER FAIRE «KX XCOR
 FAIRE «KCOR YCOR DIRE :VILLE FAIRE «CX
 XCOR FAIRE «CY YCOR
 FAIRE «SX (:DX + :KX) / 2
 FAIRE «SY (:DY + :KY) / 2
 FAIRE «VX (:DY - :KY)
 FAIRE «VY (:KX - :DX)
 FAIRE «FACT (:VX ★ (:CD - :SX) + :VY ★
 (:CY - :SY) / (:VX ★ :VX)
 (:VX ★ :VX) + (:VY ★ :VY) ★ (:VY ★
 :VY)
 FAIRE «X :SX + FACT ★ :VX
 FAIRE «Y :SY + FACT ★ :VY
 DIRE :CHEVALIER DIRIGER VERS :X :Y
 AV 10
 FIN

Déplacement amélioré

Une trajectoire améliorée pour le chevalier, qui vise à couper la route du dragon.

POUR DÉPLACEMENT.CHEVALIER
 DIRE :DRAGON FAIRE «X XCOR FAIRE «Y YCOR
 DIRE :VILLE
 FAIRE «ALLANT VERS :X :Y
 DIRE :CHEVALIER DIRIGER 270 + :ALLANT
 SI XCOR < :X ALORS GAUCHE 180
 AV 10
 FIN

Contrôle au clavier

Contrôle du dragon au clavier :

POUR DÉPLACEMENT.DRAGON
 DIRE :DRAGON DÉPLACEMENT LIREFRAPPÉ AV 10
 FIN

POUR DÉPLACEMENT :DIR
 SI :DIR = «W ALORS DIRIGER 0
 SI :DIR = «S ALORS DIRIGER 90
 SI :DIR = «Z ALORS DIRIGER 180
 SI :DIR = «A ALORS DIRIGER 270
 FIN

POUR LIREFRAPPÉ
 SI RC? ALORS SORTIR CARACTÉREL
 SORTIR»
 FIN



Nuances logo

LOGO sur le Spectrum et sur Apple ne comporte pas de figures graphiques. Les utilisateurs Atari noteront les différences suivantes :

1. FIGURETOUCHEE? n'existe pas. Ne mettez pas la ligne SI TOUCHE?, etc. dans JOUER et mettez la ligne suivante en tant que dernière ligne de METTRE.EN.PLACE.ÉCRAN :

LORSQUE TOUCHER :DRAGON :CHEVALIER
 (DRAGON.DÉTRUIT STOP)

2. Il n'y a pas d'équivalents de GRANDX, GRANDY, PETITX et PETITY. Contentez-vous de ne pas les mettre.
 3. Pour FOND, mettre DONNERFOND, et pour COULEURTRAIT, DONNERCOULEURTRAIT. Les codes pour les couleurs sont différents, bien sûr.
 4. De manière surprenante, VERS n'existe pas avec LOGO Atari (il figure pourtant dans les versions LCS de l'Apple et du Spectrum). Aussi, remplacez les lignes :

DIRIGER VERS :X :Y
 AV 10

dans SUIVRE et DÉPLACEMENT.CHEVALIER par :

FAIRE «FRAC 10 / 10 / (RACINECARREE ((
 XCOR - :X) ★ (XCOR - :X) + (YCOR -
 Y) ★ (YCOR - Y))
 PLACER LIST (XCOR + (:X - XCOR) ★ :FRAC)
 (YCOR + (:Y - YCOR) ★ :FRAC)



Opérations de suspension

Nous avons déjà évoqué le « manquement d'interruption ». Ce sont des messages qui interrompent la tâche que le processeur est en train d'exécuter pour lui transmettre une information. En voici le détail.

On trouve souvent des interruptions lorsqu'on doit faire des entrées au clavier. Si un programme accède directement au clavier pour obtenir le prochain caractère entré, alors toute touche appuyée pendant que le programme fait autre chose est ignorée. Même lorsque le processeur est totalement engagé dans un traitement d'entrée au clavier, il est encore possible de perdre un caractère, surtout s'il suit un caractère nécessitant un traitement spécial tel un retour chariot.

La solution est que le clavier interrompe le processeur — telles que l'accès aux disques — où une interruption peut causer une perte de données ou d'autres catastrophes. Il doit donc y avoir un mécanisme pour *masquer* les interruptions, afin que le processeur puisse les ignorer durant une opération particulièrement délicate. Dans ce cas, il est préférable de noter qu'une interruption a eu lieu, afin qu'elle puisse être traitée par la suite.

A l'inverse, si l'on a affaire à une interface disque qui est commandée par une interruption, alors ces interruptions seront prioritaires et en aucun cas masquées — ce qui donne lieu au concept d'interruption *non masquable*. Une telle interruption peut provenir d'un circuit qui détecte une baisse de la tension d'alimentation : son programme de traitement commencera immédiatement à sauvegarder la tâche en cours.

Il existe toutefois deux problèmes possibles. L'utilisateur peut avoir tapé si vite que le tampon s'est rempli trop rapidement et que le programme ne peut traiter les entrées, de sorte que le tampon a « débordé ». La solution requiert un compromis entre l'allocation d'une mémoire suffisante au tampon et l'absence de gaspillage de cet espace mémoire. Le second problème provient des utilisateurs, qui n'aiment pas que le caractère qu'ils viennent de taper n'apparaisse pas immédiatement à l'écran. Ils ont alors tendance à continuer à appuyer, générant ainsi des douzaines de caractères qui entrent dans le tampon (et à nouveau le font déborder). Le problème est généralement résolu lorsqu'on est familiarisé avec l'ordinateur.

Une autre application utile des interruptions a lieu lorsqu'une sortie est envoyée à l'imprimante. Durant l'impression, le processeur doit, par exemple, travailler pendant 100 microsecondes, lorsqu'il envoie un caractère à l'imprimante, puis il attend des milliers de microsecondes que l'imprimante ait traité ce caractère. Un système de *bobinage* peut y remédier : il place les fichiers à imprimer dans une queue, et une partie du premier fichier de la queue est chargée dans une autre zone tampon de la mémoire. Le port qui sert à l'imprimante interrompra le processeur chaque fois que l'imprimante sera prête à envoyer

un autre caractère. Le programme de traitement des interruptions envoie alors le caractère suivant du tampon ou charge la prochaine section de fichier en tête de la queue dans le tampon. De la sorte, l'impression peut se faire, tout en laissant le processeur libre de continuer autre chose.

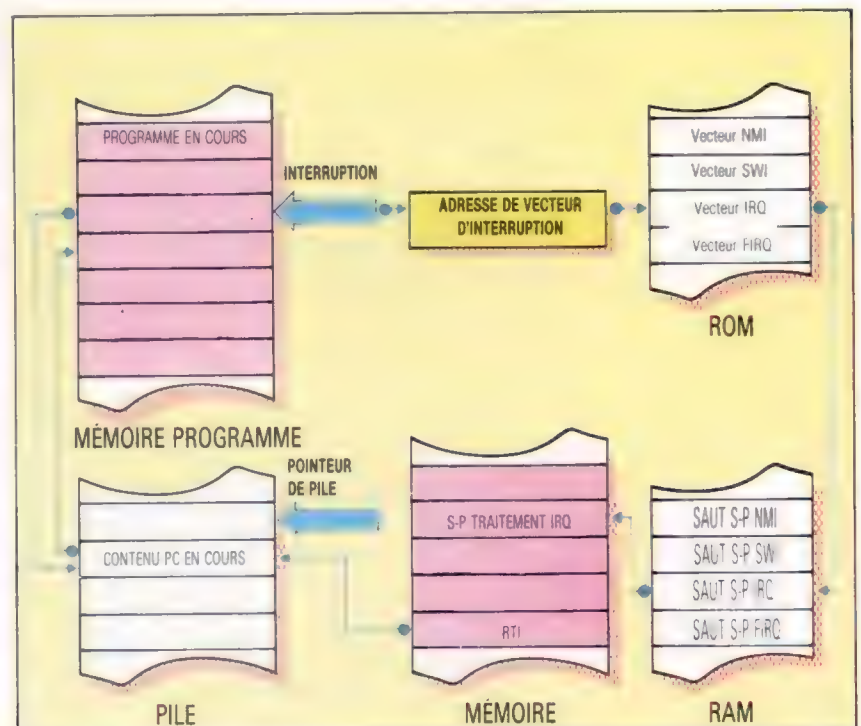
Types d'interruptions

Il y a des opérations accomplies par le processeur — telles que l'accès aux disques — où une interruption peut causer une perte de données ou d'autres catastrophes. Il doit donc y avoir un mécanisme pour *masquer* les interruptions, afin que le processeur puisse les ignorer durant une opération particulièrement délicate. Dans ce cas, il est préférable de noter qu'une interruption a eu lieu, afin qu'elle puisse être traitée par la suite.

A l'inverse, si l'on a affaire à une interface disque qui est commandée par une interruption, alors ces interruptions seront prioritaires et en aucun cas masquées — ce qui donne lieu au concept d'interruption *non masquable*. Une telle interruption peut provenir d'un circuit qui détecte une baisse de la tension d'alimentation : son programme de traitement commencera immédiatement à sauvegarder la tâche en cours.

Interruption 1

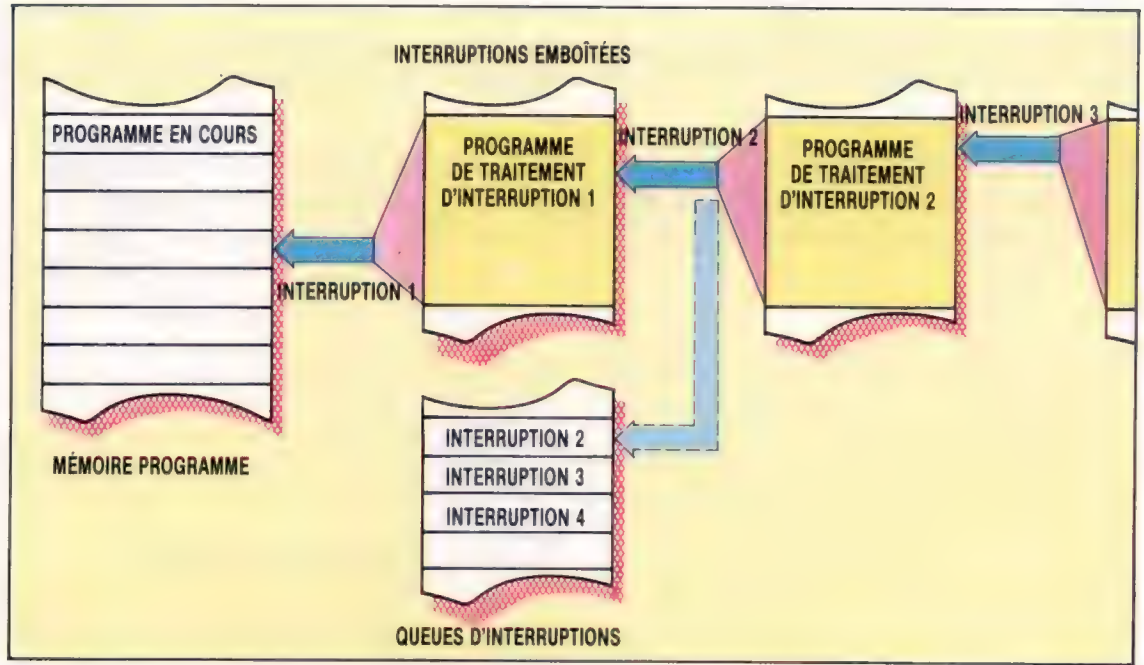
En cas d'interruption, le processeur achève le traitement de l'instruction et met sur pile le contenu du compteur de programme. L'adresse du vecteur d'interruption approprié est alors chargée dans le compteur de programme, et le contrôle passe à cette adresse — généralement en ROM. Celle-ci indique une autre adresse — généralement en RAM — où une instruction JMP dirige le contrôle au programme de traitement d'interruptions. Puisque l'instruction JMP est stockée en RAM, elle peut être trouvée et modifiée par le programmeur afin que, lors d'une interruption, le contrôle passe d'abord à une routine utilisateur spéciale avant d'aller au programme de traitement normal. (Cl. Liz Dixon.)



Interruptions interrompues

Si une interruption a lieu pendant une interruption, une solution pour le processeur est d'« emboîter » les interruptions : chaque fois qu'une interruption se produit, le PC est mis sur pile, la nouvelle interruption est traitée immédiatement, et le contrôle revient à l'adresse mise sur pile. Cet emboîtement est limité par la capacité des piles et la capacité des dispositifs générant des interruptions d'admettre des délais dans le traitement de leurs interruptions.

Une autre solution pour le processeur consiste à mettre sur pile les détails de toute interruption sur une queue d'interruption. Lorsque la première interruption est terminée, le processeur inspecte la queue, fait une interruption aussi longtemps qu'il en trouve dans la queue, et finalement repasse le contrôle au programme.



Dans le cas où des interruptions proviennent de plusieurs sources, il faut considérer la possibilité d'interruptions *emboîtées*. Si une interruption a lieu pendant que le processeur est en train d'effectuer une autre interruption, il y a deux stratégies possibles. D'abord, la nouvelle interruption peut être ignorée jusqu'à ce que celle en cours soit achevée. Ensuite, on peut classer les interruptions selon leur niveau d'urgence, de sorte qu'une interruption à haute priorité passe avant le traitement de celle à plus faible priorité.

Interruptions de logiciel

L'instruction *SWI* peut servir dans un programme comme moyen de retourner au système d'exploitation en générant sa propre interruption — appelée *interruption logiciel* (par opposition aux interruptions générées par le matériel dont nous avons parlé jusqu'ici).

Les instructions *SWI* peuvent donc servir de points d'interruption dans un programme en langage machine pour aider au débogage; cette possibilité est offerte par beaucoup de moniteurs de langage machine en ROM, ainsi que des programmes de débogage. L'utilisateur choisit des points dans le code où l'exécution du programme doit s'arrêter, et les instructions y sont remplacées par des *SWI*.

Lorsque le programme est exécuté, le programme de traitement d'interruptions permet alors au programmeur d'inspecter et éventuellement de modifier le contenu des registres et des emplacements mémoire, et de voir exactement ce que fait le programme. En reprenant l'exécution, le moniteur/débogueur remplace l'instruction affichée par le point d'interruption *SWI*, et continue le programme à partir de ce point.

Le 6809 a trois différents mécanismes d'interruption : *IRQ* (Interrupt ReQuest, demande d'interruption), *FIRQ* (First Interrupt ReQuest, première

demande d'interruption) et *NMI* (Non-Maskable Interrupt, interruption non masquable). Elles sont toutes activées par le signal approprié qui est reçu sur trois contacts de la puce de processeur. La barre au-dessus du nom (dans *IRQ*, par exemple) indique qu'elles sont activées par un signal nul au processeur, plutôt que par un 1. Ces trois contacts sont connectés au bus principal, de sorte que les puces périphériques telles que 6820 et 6850 peuvent avoir leurs contacts de demande d'interruption connectés aux mêmes lignes de bus. Lorsque les puces sont programmées, les interruptions peuvent être validées, et alors les signaux appropriés seront automatiquement émis.

Il existe aussi des interruptions logiciel causées par les instructions *SWI*, *SWI2* et *SWI3*.

En cas d'interruption, le contrôle passe à l'adresse *vecteur* contenue dans un emplacement spécifique en haut de la mémoire. Ces adresses vecteurs se trouvent généralement en ROM, de sorte que le contrôle passera toujours de là à la même adresse fixée. Toutefois, cette adresse se trouvera normalement en RAM et contiendra une instruction *JMP*, de sorte que la destination finale peut être modifiée pour le programme de traitement de l'utilisateur. Voici les emplacements mémoires :

Type d'interruption	Vecteur
NMI	\$FFFC
SWI	\$FFFA
IRQ	\$FFF8
FIRQ	\$FFF6
SWI2	\$FFF4
SWI3	\$FFF2

Il faut aussi noter que les deux octets supérieurs de la mémoire — \$FFFE et \$FFFF — contiennent le



vecteur de réinitialisation, l'adresse à laquelle est transféré le contrôle lors d'un débranchement ou d'une remise à zéro matérielle; c'est généralement l'adresse de début de moniteur ROM.

L'information concernant les interruptions est contenue dans trois bits du registre de code condition (CC) : le bit 4 (I), le bit 6 (F) et le bit 7 (E). si l'on met le bit I à 1 on masque l'interruption \overline{IRQ} , le bit F à 1 masque \overline{FIRQ} . Le bit E est utilisé par le processeur pour distinguer entre \overline{IRQ} , NMI et \overline{FIRQ} : si E est mis à 1, il a eu \overline{IRQ} ou NMI; s'il est à zéro, c'est \overline{FIRQ} .

Lorsqu'une interruption est reçue, elle est traitée d'une façon semblable à un appel de sous-programme : le contenu de certains registres peut retourner au même point du programme en cours d'exécution. Le programme de traitement d'interruptions se termine par une instruction RTI, qui ressort les registres de la pile et rend le contrôle au programme initial.

La vraie différence entre \overline{FIRQ} et les deux autres interruptions est que \overline{FIRQ} ne met sur pile que les registres compteur de programme (PC) et code condition (CC), et c'est pourquoi il est beaucoup plus rapide que les deux autres. Le programme de traitement d'interruptions doit cependant restaurer tous les registres qu'il utilise, donc ce type d'interruption ne doit pas servir pour des routines qui utilisent plus d'un ou deux registres. Nous voyons maintenant où est utilisé le bit E, parce que la même instruction RTI sert à terminer les routines \overline{IRQ} et \overline{FIRQ} , mais le processeur doit déterminer quel registre doit être sorti de la pile. La séquence d'événements lorsqu'une interruption a lieu est la suivante :

1. L'instruction en cours est exécutée.
2. Le bit I est mis, invalidant d'autres interruptions \overline{IRQ} . Si l'interruption était \overline{FIRQ} ou NMI, alors le bit F est également mis pour invalider \overline{FIRQ} . SW12 et SW13 ne masquent pas d'autres interruptions, mais SW1 le fait.
3. Lors d'un \overline{FIRQ} , le bit E est mis à zéro, sinon à 1.
4. Le vecteur dans l'emplacement mémoire approprié est chargé dans PC et l'exécution continue à partir de cette adresse.

Notre premier programme considère un autre bon usage des interruptions; il s'agit de maintenir une horloge en temps réel. Nous supposons qu'un dispositif d'horloge, qui peut être une puce spéciale comme le rythmeur 6840 ou une division du système d'horloge, ou bien une modification de l'alimentation 50 Hz est connecté à un PIA en \$E000. Le premier sous-programme permettra les interruptions et mettra un compteur 16 bits en \$50. Le programme de traitement d'interruptions incrémente simplement le compteur afin que l'inspection de \$50 donne toujours le nombre de signaux de synchronisation qui ont été reçus, à partir duquel le temps peut être calculé si le moment de départ et la fréquence des signaux de synchronisation sont connus.

Le second programme suppose qu'une imprimante est connectée au même PIA en \$E000. Nous emploierons un tampon, de longueur indéterminée, en \$100 pour stocker une ligne en sor-

tie à imprimer par cette routine. Un drapeau en \$50 est mis à zéro lorsque la ligne s'imprime, et à 1 lorsqu'elle est finie. Cela permettra de remplir le tampon avec d'autres routines. Les emplacements \$51 et \$52 contiennent un pointeur de tampon donnant l'adresse du prochain caractère à imprimer. Le premier sous-programme met le PIA, le drapeau et le pointeur de tampon pour une nouvelle ligne.

Programme 1

PIACR	EQU	\$E001	Registre de contrôle PIA.
PIADR	EQU	\$E000	Registre de données PIA.
INTRP	EQU	\$2000	
CLK1	EQU	\$50	
CLK2	EQU	\$51	
	ORG	\$1000	Sous-programme pour initialiser l'horloge.
INITCK	CLR	CLK1	Met à zéro emplacements d'horloge.
	CLR	CLK2	
	LDA	##%00000101	Valide interruptions PIA.
	STA	PIACR	
	ANDCC	##%11101111	Valide \overline{IRQ} .
WAITCK	TST	CLK2	Attend la première incrémentation.
	BEQ	WAITCK	
	RTS		
	ORG		
	LDA	INTRP	Programme de traitement d'interruption.
	LDD	PIADA	
	ADD	CLK1	Efface interruption.
	STD	#1	Prend le compte.
	RTI	CLK1	Incrémente le compte.

Programme 2

PIACR	EQU	\$E001	Registre de contrôle PIA.
PIADR	EQU	\$E000	Registre de données PIA.
INTRP	EQU	\$2000	
CR	EQU	13	Retour de chariot.
BUFFER	EQU	\$100	Adresse de tampon.
			Pointeur de tampon.
BUFPTR	EQU	\$51	Drapeau fin-de-ligne.
FLAG	EQU	\$50	Sous-programme pour tout initialiser.
	ORG	\$1000	Met à zéro le drapeau fin-de-ligne.
	CLR	FLAG	Initialise le pointeur de tampon au début du tampon.
	LDX	#BUFFER	
	STX	BUFPTR	Registre de direction de donnée d'adresse.
	CLR	PIACR	Met toutes les lignes en sortie.
	LDA	\$FF	
	STA	PIADR	Valide interruptions PIA.
	LDA	##%00000101	
	STA	PIACR	Valide \overline{IRQ} .
	ANDCC	##%11101111	
	RTS		
	ORG		
	INTRP		Programme de traitement d'interruptions.
	LDX	BUFPTR	Pointeur de tampon.
	LDA	,X+	Prend caractère suivant du tampon.
	STA	PIADR	L'imprime.
	LDB	PIADR	Efface interruption.
	STX	BUFPTR	Incrémente pointeur de tampon.
			Était-ce une fin-de-ligne?
	CMPA	#CR	Sauter si ce n'est pas une fin-de-ligne.
	BNE	FINISH	
	INC	FLAG	Sinon mettre drapeau.
FINISH	RTI		



La carte Z80 (suite)

Voici reproduits, avec l'aimable autorisation de Zilog Inc., de nouveaux éléments de la carte référence du programmeur Z80. Pour tous ceux qui veulent aller plus loin dans la programmation...

Groupe arithmétique 16 bits

		SOURCE					
		BC	DE	HL	SP	IX	IY
« AJOUTE »	HL	09	19	29	39		
	IX	DD 09	DD 19		DD 39	DD 29	
	IY	FD 09	FD 19		FD 39		FD 29
AJOUTE AVEC RETENUE ET MET DRAPEAUX « ADC »		HL	ED 4A	ED 5A	ED 6A	ED 7A	
SOUSTRAIT AVEC RETENUE ET MET DRAPEAUX « SBC »		HL	ED 42	ED 52	ED 62	ED 72	
INCREMENTE « INC »			03	13	23	33	DD 23 FD 23
DÉCRÉMENTE « DEC »			08	1B	2B	3B	DD 2B FD 2B

Mnémonique	Opération symbolique	S	Z	Drapeaux H			P/V	N	C	76	Opc	543	210	Hex	Nombre d'octets	Nombre de cycles M	Nombre d'états T	Commentaires
ADD HL,ss	HL ← HL + ss	•	•	X	X	X	•	0	‡	00	ss1	001			1	3	11	ss Reg. 00
ADC HL, ss	HL ← HL + ss + CY	‡	‡	X	X	X	V	0	‡	11 01	101 ss1	101 010		ED	2	4	15	01 DE 10 HL 11 SP
SBC HL, ss	HL ← HL - ss - CY	‡	‡	X	X	X	V	1	‡	11 01	101 ss0	101 010		ED	2	4	15	
ADD IX, pp	IX ← IX + pp	•	•	X	X	X	•	0	‡	11 01	011 pp1	101 0001		DD	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	IY ← IY + rr	•	•	X	X	X	•	0	‡	11 00	111 rr1	101 001		FD	2	4	15	rr Reg. 00 BC 01 DE 10 IY 11 SP
INC ss	ss ← ss + 1	•	•	X	•	X	•	•	•	00	ss0	011			1	1	6	
INC IX	IX ← IX + 1	•	•	X	•	X	•	•	•	11 00	011 100	101 011		DD 23	2	2	10	
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	•	11 00	111 100	101 011		FD 23	2	2	10	
DEC ss	ss ← ss - 1	•	•	X	•	X	•	•	•	00	ss1	011			1	1	6	
DEC IX	IX ← IX - 1	•	•	X	•	X	•	•	•	11 00	011 101	101 011		DD 2B	2	2	10	
DEC IY	IY ← IY - 1	•	•	X	•	X	•	•	•	11 00	111 101	101 011		FD 2B	2	2	10	

Notes : dd est l'une quelconque des paires de registres BC, DE, HL, SP;
qq est l'une quelconque des paires de registres AF, BC, DE, HL;
rr est l'une quelconque des paires de registres BC, DE, IY, SP.

Notation de drapeaux : • = drapeau non affecté, 0 = drapeau remis, 1. 1 = drapeau mis, X = drapeau inconnu;
‡ = drapeau affecté selon le résultat de l'opération.

**Page manquante
(publicité)**

**Page manquante
(publicité)**